



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Química

FELIPE QUEIROZ MENDES

MODELAGEM E SIMULAÇÃO DE SISTEMAS GÁS-SÓLIDO
POLIDISPERSOS: CICLONES COM ALTA CARGA

MODELING AND SIMULATION OF POLYDISPERSE GAS-SOLID
SYSTEMS: HIGH-LOADED CYCLONES

CAMPINAS

2019



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Química

FELIPE QUEIROZ MENDES

MODELAGEM E SIMULAÇÃO DE SISTEMAS GÁS-SÓLIDO
POLIDISPERSOS: CICLONES COM ALTA CARGA

Dissertação apresentada à Faculdade de Engenharia Química da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Química

Orientador: PROF. DR. DIRCEU NORILER

ESTE EXEMPLAR CORRESPONDE À
VERSÃO FINAL DA DISSERTAÇÃO DE-
FENDIDA PELO ALUNO FELIPE QUEI-
ROZ MENDES, E ORIENTADA PELO
PROF. DR. DIRCEU NORILER

PROF. DR. DIRCEU NORILER

CAMPINAS

2019

Agência(s) de fomento e nº(s) de processo(s): CNPq, 132481/2017-0

ORCID: <https://orcid.org/0000-0001-8593-7537>

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Elizângela Aparecida dos Santos Souza - CRB 8/8098

Mendes, Felipe Queiroz, 1991-
M522m Modelagem e simulação de sistemas gás-sólido polidispersos : ciclones com alta carga / Felipe Queiroz Mendes. – Campinas, SP : [s.n.], 2019.

Orientador: Dirceu Noriler.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Modelagem e simulação. 2. Ciclones. 3. Escoamento multifásico. 4. Modelo de balanço populacional. 5. Fluidodinâmica Computacional. I. Noriler, Dirceu, 1978-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Química. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Modeling and simulation of polydisperse gas-solid systems : high-loaded cyclones

Palavras-chave em inglês:

Modeling and simulation

Cyclones

Multiphase flow

Population balance model

Computational fluid dynamics

Área de concentração: Engenharia Química

Titulação: Mestre em Engenharia Química

Banca examinadora:

Dirceu Noriler [Orientador]

Marcelo Souza de Castro

Francisco José de Souza

Data de defesa: 15-02-2019

Programa de Pós-Graduação: Engenharia Química

Dissertação de Mestrado defendida por Felipe Queiroz Mendes e aprovada em
15 de fevereiro de 2019 pela banca examinadora constituída pelos doutores:

Prof. Dr. Dirceu Noriler - Orientador
FEQ/UNICAMP

Prof. Dr. Marcelo Souza de Castro
FEM/UNICAMP

Prof. Dr. Francisco José de Souza
Universidade Federal de Uberlândia - UFU

A Ata de defesa com as respectivas assinaturas dos membros encontra-se no
SIGA/Sistema de Fluxo de Tese e na Secretaria do Programa da Unidade.

*Dedico esta dissertação aos meus pais, Manoel e Luzia, que sempre deram suporte para
que todos os filhos estudassem e buscassem o seu caminho.*

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro (Número do processo: 132481/2017-0).

À UNICAMP, em especial a Faculdade de Engenharia Química em nome todos seus funcionários e colaboradores, pela oportunidade de fazer o mestrado numa das melhores universidades do Brasil.

Ao Prof. Dr. Dirceu Noriler por toda orientação durante a trajetória, me ensinando tanto durante o percurso desta caminhada.

Ao Prof. Dr. Sávio Souza Venâncio Vianna por me supervisionar durante um ano no Programa de Estágio Docente - PED.

A todos do Laboratório de Pesquisa em Processos Químicos e Gestão Empresarial - PQGe pela companhia, pelas conversas, pelos conselhos, pelos cafés, por tudo que me deu forças para executar este trabalho.

À minha família, meus pais, Luzia e Manoel, meus irmãos e seus cônjuges, Fabrícia e André, Flávia e Marcos, Fábio e Crys e meus sobrinhos, Mateus, Artur e Davi, que sempre estão ao meu lado me apoiando.

À minha namorada, Daniella Flávia Schettini, por estar sempre me apoiando e incentivando até mesmo nos momentos mais difíceis e também seus pais, João Flávio e Enézia Maria, por me darem muito carinho e apoio.

A todos aqueles que de alguma forma contribuíram para a realização deste trabalho e aqueles que passaram pela minha vida e de alguma forma moldaram o que sou hoje

*“What we do for ourselves dies with us. What we do for others and the world remains
and is immortal.”*

Albert Pike

RESUMO

Ciclones são dispositivos amplamente utilizados na indústria para separação de sólidos a partir de correntes gasosas devido a suas características de fácil construção, baixo custo de operação e manutenção, alta eficiência de separação para partículas maiores que $5\ \mu m$, além da capacidade de suportar altas temperaturas e pressões. Em consequência disto, eles são extensivamente estudados no intuito de melhorar sua eficiência de coleta com menor custo energético possível (menor queda de pressão). Apesar de sua simples construção e operação, o comportamento fluidodinâmico nesses sistemas é complexo, de modo que a técnica de fluidodinâmica computacional (CFD), aliada aos estudos experimentais tradicionalmente conduzidos, apresenta-se como uma alternativa viável para compreensão e consequente projeto e otimização de suas condições geométricas e operacionais. No intuito de melhor caracterizar o comportamento dos ciclones, curvas de eficiência devem ser obtidas para cada geometria e condição operacional. Assim, um modelo fluidodinâmico baseado nas ferramentas de CFD pode contribuir muito com o tempo e custo de projeto e otimização destes dispositivos. No entanto, a literatura não apresenta um modelo consolidado que disponha de precisão na predição de curvas de eficiência de ciclones de alta carga em tempo computacional razoável, a despeito do grande avanço de *hardware* obtidos nos últimos anos. Diante disso, neste trabalho foi proposta a aplicação da abordagem Euleriana-Euleriana acoplada às equações de balanço populacional, resolvida através do método de quadratura direta baseado em momentos (DQMOM), para modelagem de um ciclone com alta carga de sólidos. Para a solução das equações do modelo, foi utilizado o código comercial Fluent 17.2, de modo que as equações de balanço populacional resolvidas pelo método DQMOM foram implementadas em linguagem C através de *User Define Functions* (UDF) para o acoplamento destas equações ao *software*. A verificação do modelo implementado foi realizada utilizando o trabalho de Niemi (2012) que estudou um leito fluidizado pseudo-bidimensional. Todas as simulações polidispersas do ciclone com alta carga utilizando diferentes números de pontos de quadratura apresentaram perfis fluidodinâmicos similares e eficiência global próxima à experimental. Ademais, a simulação monodispersa utilizando a abordagem Euleriana-Euleriana tradicional obteve uma eficiência muito mais baixa, mostrando a viabilidade da metodologia proposta. Por fim, a curva de distribuição de tamanho de partículas coletadas pelo ciclone foram reconstruídas utilizando o método de máxima entropia a fim de determinar a curva de eficiência de coleta. Foi observado que com o aumento do número de quadraturas houve uma convergência em direção à curva experimental, de modo que para quatro, cinco e seis pontos de quadratura foi obtido um erro relativo menor que 5% comparado aos dados experimentais.

Palavras-chaves: Abordagem Euleriana-Euleriana; Balanço Populacional; Fluidodinâmica Computacional.

ABSTRACT

Cyclones are devices widely used in industries to separate solids from gas streams due to their easy construction characteristics, low operation and maintenance costs, high separation efficiency for particles larger than $5\text{ }\mu\text{m}$ and ability to withstand high temperatures and pressures. As a result of these advantages, cyclones are widely studied in order to improve the collection efficiency with lowest energy cost possible (lower pressure drop). In contrast to the characteristics of simple construction and operation, the fluid dynamic behavior in these systems is quite complex so that the computational fluid dynamics (CFD) technique, together with the experimental studies traditionally conducted, is a viable alternative in the understanding, and consequently design and optimization of their geometric and operational conditions. In order to better characterize the behavior of cyclones, efficiency curves should be obtained for each geometry and operational condition. Thus, a fluid-dynamic model based on CFD can contribute with time and cost of design and optimization of these devices. However, the literature does not present a consolidated model that presents precision in the prediction of efficiency curves of high load cyclones in reasonable computational time, in spite of the great advance of hardware obtained in recent years. Therefore, in this work, an Eulerian-Eulerian approach coupled to the population balance equations solved by the direct quadrature method of moments (DQ-MOM) for the simulation of a cyclone with high particle mass loading was proposed. For the solution of the equations of the model, the commercial code Fluent 17.2 was used, so the population balance equations solved by the DQMOM method were implemented in C language through User Define Functions (UDF) for the coupling of these equations to the software. The verification of the implemented model was done using the work of Niemi (2012) who studied a pseudo-two-dimensional fluidized bed. All the cyclone simulations using different numbers of quadrature points presented similar fluid dynamics profiles and global efficiency close to the experimental one. Moreover, the monodisperse simulation using the traditional Eulerian-Eulerian approach with only the mean diameter obtained a much lower efficiency, showing the feasibility of the methodology proposed by this study. Finally, the efficiency curve was calculated using the maximum entropy method to reconstruct the particle size distribution at the outlet of the cyclone to the hopper. The increase in the number of quadrature showed a convergence toward the experimental curve, so that the relative error obtained was less than 5% for four, five and six quadrature points compared to the experimental data.

Keywords: Eulerian-Eulerian Approach; Population Balances; Computational Fluid Dynamics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ciclones na indústria de produção de cimento	22
Figura 2 – Esquema da geometria de um ciclone mostrando os oito parâmetros principais	26
Figura 3 – Tipos de entradas em ciclone	27
Figura 4 – Características fenomenológicas de um ciclone	29
Figura 5 – Perfil da velocidade tangencial	30
Figura 6 – Curva típica de eficiência de coleta de ciclones	32
Figura 7 – Exemplo de uma função de densidade de número e de massa para uma determinada amostra de partículas	33
Figura 8 – Esquema da adição de 15 tubos com 6 <i>mm</i> de diâmetro e 20 <i>mm</i> de comprimento na entrada tangencial do ciclone	39
Figura 9 – Esquema das três variações propostas por Wasilewski e Brar (2017) . .	40
Figura 10 – Comparação da curva de eficiência experimental e numérica obtida por Huang <i>et al.</i> (2018) para o ciclone com carga de sólidos de 115,3 <i>g/m</i> ³	42
Figura 11 – Exemplo da correção da fração volumétrica	54
Figura 12 – Fluxograma de execução do processos do Fluent para o solver baseado na pressão em cada passo de tempo	56
Figura 13 – Esquema da geometria do CFB em <i>cm</i> ressaltando as alturas de 23 <i>cm</i> , 40 <i>cm</i> , 80 <i>cm</i> e 120 <i>cm</i>	57
Figura 14 – Perfil de velocidade do gás utilizado como condição de contorno	59
Figura 15 – Distribuição do tamanho de partícula utilizada para avaliar a influência do número de pontos de quadraturas	60
Figura 16 – Esquema das dimensões da geometria do ciclone em milímetros	67
Figura 17 – Distribuição acumulada do tamanho de partículas experimental em massa apresentada por Huang <i>et al.</i> (2018)	68
Figura 18 – Função densidade de probabilidade de número utilizada para simulação do ciclone com alta carga de sólidos	70
Figura 19 – Sub-rotina para definir o termo convectivo das equações de transporte resultantes da aplicação do DQMOM na PBE	73
Figura 20 – Perfil da fração volumétrica média da fase sólida em diferentes alturas do leito	75
Figura 21 – Perfil da velocidade vertical média em diferentes alturas do leito	76
Figura 22 – Média transversal do diâmetro médio de Sauter em função da altura do CFB	77
Figura 23 – Mapas da fração volumétrica média no leito fluidizado circulante	78

Figura 24 – Reconstrução da distribuição do tamanho de partículas inicial utilizando os métodos máxima entropia e <i>spline</i> cúbica e diferentes números de momentos	79
Figura 25 – Comparação dos mapas da fração volumétrica média no CFB para a simulação monodispersa e polidispersa com diferentes pontos de quadratura	81
Figura 26 – Média transversal dos momentos da função densidade de probabilidade de número ao longo do leito fluidizado circulante para os diferentes pontos de quadraturas	82
Figura 27 – Média transversal dos diâmetro médio de Sauter e medidas de dispersão, assimetria e forma da função densidade de probabilidade de número ao longo do CFB para diferentes pontos de quadratura	83
Figura 28 – Reconstrução da função densidade de probabilidade de número na saída do CFB para os diferentes pontos de quadratura	84
Figura 29 – Queda de pressão pelo inverso do números de células das malhas analisadas e para a extrapolação do método GCI	85
Figura 30 – Eficiência de coleta de sólidos pelo inverso do números de células das diferentes malhas analisadas e para a extrapolação do método GCI	86
Figura 31 – Incerteza numérica do perfil da velocidade tangencial média da malha refinada calculada utilizando método do GCI na altura de 216 <i>mm</i> logo abaixo da entrada do <i>vortex finder</i>	87
Figura 32 – Mapas da fração volumétrica média da fase sólida no ciclone para as simulações monodispersa e polidispersa com diferentes números de pontos de quadratura	88
Figura 33 – Velocidade tangencial média da fase gasosa no ciclone com alta carga de sólidos	89
Figura 34 – Velocidade tangencial média pela posição radial do ciclone na altura de 216 <i>mm</i> comparando diferentes simulações	90
Figura 35 – Comparação do perfil médio do diâmetro médio de Sauter no ciclone para diferentes pontos de quadratura	91
Figura 36 – Perfis do diâmetro médio de Sauter na altura de 216 <i>mm</i> das simulações polidispersas com diferentes números de pontos de quadratura	91
Figura 37 – Reconstrução da distribuição mássica do tamanho de partículas na saída do ciclone para o <i>hopper</i> dos modelos polidispersos com diferentes números de pontos de quadratura	92
Figura 38 – Curva de eficiência do ciclone de alta carga considerando as partículas coletadas utilizando o método da máxima entropia para reconstrução da PDF para os diferentes números de pontos de quadratura	93

Figura 39 – Comparação da curva de eficiência do ciclone de alta carga com a simulação e dados experimentais de Huang <i>et al.</i> (2018)	94
Figura 40 – Reconstrução da distribuição do tamanho de partículas em massa para o arraste de duas e quatro vias	95
Figura 41 – Curva de eficiência do ciclone de alta carga considerando as partículas coletadas utilizando o método da máxima entropia para reconstrução da PDF para os modelos de arraste de duas e quatro vias	96
Figura 42 – Perfis da velocidade tangencial média da fase gasosa na altura de 216 <i>mm</i> para os casos 1 e 2	97
Figura 43 – Curva de eficiência de coleta para os casos: (a) 1; (b) 2; (c) 3 e (d) 4 .	98

LISTA DE TABELAS

Tabela 1 – Parâmetros geométricos dos modelos de ciclones mais conhecidos . . .	27
Tabela 2 – Modelos empíricos para o calculo da queda de pressão	31
Tabela 3 – Modelos mais comuns para ajuste de distribuições do tamanho de partículas	35
Tabela 4 – Síntese dos trabalhos avaliados na revisão bibliográfica	43
Tabela 5 – Síntese das equações básicas da modelagem matemática que serão resolvidas na simulação numérica de ciclones com alta carga de sólidos .	55
Tabela 6 – Modelos e parâmetros utilizados na modelagem de Niemi (2012) . . .	58
Tabela 7 – Esquemas e algoritmos numéricos utilizados na simulação realizada por Niemi (2012)	59
Tabela 8 – Momentos da distribuição utilizada para avaliar a influência do número de pontos de quadraturas	61
Tabela 9 – Os pesos (w), as abcissas (ξ) e as frações volumétricas (α) iniciais do CFB de cada ponto de quadratura (q) para os casos com o total de N pontos de quadraturas	61
Tabela 10 – Esquemas e algoritmos numéricos utilizados na simulação do teste de malha para o ciclone com alta carga	70
Tabela 11 – Momentos da função densidade de probabilidade de número utilizada para simulação do ciclone com alta carga de sólidos	71
Tabela 12 – Os pesos (w), as abcissas (ξ) e as frações volumétricas (α) de cada ponto de quadratura (q) para os casos com o total de N pontos de quadraturas da condição de contorno de entrada do ciclone de alta carga	71
Tabela 13 – Velocidade, carga de sólidos e fração volumétrica da entrada do ciclone	72
Tabela 14 – Funções utilizadas para obter o valor de variáveis em cada célula da malha	74
Tabela 15 – Erro relativo das reconstruções utilizando os métodos de máxima entropia e <i>spline</i> cúbica	80
Tabela 16 – Erro relativo entre as quadraturas para os momentos de ordem 1, 2, 3 e 4	81
Tabela 17 – Tempo de simulação para os diferentes números de pontos de quadratura	82
Tabela 18 – Comparação da queda de pressão e eficiência global entre diferentes simulações	87
Tabela 19 – Erro relativo das curvas de eficiência para os diferentes pontos de quadraturas comparado com o resultado experimental de Huang <i>et al.</i> (2018)	93

Tabela 20 – Resultados da queda de pressão e eficiência global de coleta das simulações dos casos apresentados na seção 3.2.6.4 e os resultados experimentais da eficiência global de coleta obtidos por Huang *et al.* (2018) . 97

LISTA DE ACRÔNIMOS, ABREVIACÕES E SÍMBOLOS

Acrônimos

CFB	Leito Fluidizado Circulante - <i>Circulating Fluidized Bed</i>
CFD	Fluidodinâmica Computacional - <i>Computational Fluid Dynamics</i>
DQMOM	Método de Quadratura Direta Baseado em Momentos - <i>Direct Quadrature Method of Moments</i>
FCC	Craqueamento Catalítico Fracionado - <i>Fluid Catalytic Cracking</i>
GCI	Índice de Convergência de Grade - <i>Grid Convergence Index</i>
NDF	Função Densidade de Número - <i>Number Density Function</i>
PBE	Equações de Balanço Populacional - <i>Population Balance Equation</i>
PDF	Função Densidade de Probabilidade - <i>Probability Density Function</i>
PIV	Velocimetria por Imagem de Partículas - <i>Particle Image Velocimetry</i>
PSD	Distribuição do Tamanho de Partícula - <i>Particle Size Distribution</i>
QMOM	Método de Quadratura Baseado em Momentos - <i>Quadrature Method of Moments</i>
RANS	Simulação numérica de escoamentos turbulentos via equações de médias de Reynolds - <i>Reynolds Averaged Navier-Stokes</i>
RSM	Modelos de Tensões de Reynolds - <i>Reynolds Stress Model</i>
SMD	Diâmetro Médio de Sauter - <i>Sauter Mean Diameter</i>
UDF	Funções Definidas pelo Usuário - <i>User Define Function</i>
UDM	Memoria Definida pelo Usuário - <i>User-Defined Memory</i>
UDS	Escalar Definido pelo Usuário - <i>User-Defined Scalar</i>

Letras Gregas

ϵ	Termo de dissipação
------------	---------------------

λ	Multiplicadores de Lagrange	
ξ	Coordenadas internas do Balanço Populacional	
Π	Termo de deformação devido a tensão	
τ	Tensão	$[M \cdot L^{-1} \cdot T^{-2}]$
Ψ	Termo da taxa de produção de tensão turbulenta	
α	Fração volumétrica	
β	Vetor do algoritmo produto-diferença	
δ	Função delta de Kronecker	
ε	Taxa de dissipação da energia cinética turbulenta	
η	Eficiência	
λ	Viscosidade volumétrica - <i>Bulk viscosity</i>	$[M \cdot L^{-1} \cdot T^{-1}]$
μ	Viscosidade dinâmica - <i>Shear viscosity</i>	$[M \cdot L^{-1} \cdot T^{-1}]$
ξ	Abcissa da quadratura - representa o diâmetro da quadratura	$[L]$
ρ	Massa específica	$[M \cdot L^{-3}]$
σ	Desvio Padrão	$[L]$
ϕ	Variável de interesse para o estudo	
Ω	Velocidade angular	$[T^{-1}]$

Letras Latinas

D_M	Termo da difusão molecular	
D_T	Termo da difusão turbulenta	
g	Aceleração da gravidade	$[L \cdot T^{-2}]$
I	Tensor identidade	
v	Velocidade	$[L \cdot T^{-1}]$
x	Coordenadas externas ou espaciais do Balanço Populacional	
Y	Características da fase contínua que afetam o comportamento das partículas	
a	Diagonal principal da matriz simétrica tridiagonal do algoritmo produto-diferença	

b	Co-diagonal da matriz simétrica tridiagonal do algoritmo produto-diferença	
C	Constante	
c	Número total de células da malha	
C_{fr}	Coefficiente de Fricção	
d	Diâmetro	$[L]$
E	Erro relativo	
e	Coefficiente de Restituição	
Eu	Número de Euler	
F	Fator de Segurança do GCI	
f	Função de Densidade de Probabilidade	
g_0	Função de Distribuição Radial	
h	Função que representa o nascimento e morte das partículas	
I	Parâmetros da <i>spline</i> cúbica	
K	Coefficiente de arrasto entre duas fases	
k	Energia cinética turbulenta	
L	Dimensões da geometria do ciclone	$[L]$
l	Tamanho da malha	$[L]$
M	Momento	
m	Massa	$[M]$
N	Número total de pontos de quadratura	
n	Número de subintervalos	
o	Ordem Aparente do GCI	
P	Primeira matriz do algoritmo produto-diferença	
p	Pressão	$[M \cdot L^{-1} \cdot T^{-2}]$
R	Fator de Refinamento entre Duas Malhas	
r	Raio	$[L]$

Re	Número de Reynolds	
s	Função polinomial de terceiro grau	
T	Matriz simétrica tridiagonal do algoritmo produto-diferença	
t	Tempo	$[T]$
V	Volume	$[L^3]$
w	Peso da quadratura	

Sobreescrito

'	Primeira derivada ou Flutuação
"	Segunda derivada
·	Taxa
–	Média
~	Valor não correto
C	Central
ext	Extrapolado

Subescrito

1	Malha refinada
2	Malha intermediária
3	Malha grosseira
a	Altura da entrada do escoamento gás-sólido
b	Largura da entrada do escoamento gás-sólido
c	Ciclone ou seção cilíndrica do ciclone
d	Seção cônica do ciclone
E	Modelo de arraste de Ergun (1952)
e	<i>Vortex finder</i>
g	Fase gasosa
k	Ordem do momento

p	Ponto de quadratura
q	Ponto de quadratura
R	Reynolds
s	Fase sólida ou particulada
t	Turbulento
WY	Modelo de arraste de Wen e Yu (1966)

SUMÁRIO

1	INTRODUÇÃO	22
1.1	MOTIVAÇÃO PELO TEMA	23
1.2	PROBLEMA, HIPÓTESES E QUESTIONAMENTOS	23
1.3	OBJETIVOS GERAIS	24
1.3.1	OBJETIVOS ESPECÍFICOS	24
1.4	ESTRUTURA DA DISSERTAÇÃO	25
2	FUNDAMENTAÇÃO TEÓRICA	26
2.1	CICLONES	26
2.1.1	Fenomenologia	28
2.1.2	Parâmetros de desempenho	30
2.1.2.1	Queda de pressão	30
2.1.2.2	Eficiência de coleta	31
2.2	DISTRIBUIÇÃO DO TAMANHO DE PARTÍCULAS	33
2.3	REVISÃO BIBLIOGRÁFICA	35
2.3.1	Abordagem Euleriana-Euleriana	36
2.3.2	Abordagem Euleriana-Lagrangeana	37
2.4	PONTO DE INVESTIGAÇÃO	42
3	METODOLOGIA	44
3.1	MODELAGEM MATEMÁTICA	44
3.1.1	Desenvolvimento do Modelo Euleriano-Euleriano	44
3.1.2	Modelo de Coeficiente de Arraste	46
3.1.3	Modelo de Diâmetro Característico	46
3.1.4	Modelo de Turbulência	47
3.1.5	Equações de Balanço Populacional	48
3.1.5.1	Método de Quadratura Direta Baseado em Momentos	50
3.1.5.2	Abordagem de Mistura	51
3.1.5.3	Correção da Fração Volumétrica	53
3.1.6	Síntese da Modelagem Matemática	54
3.2	SIMULAÇÃO NUMÉRICA	55
3.2.1	Implementação das Equações do Modelo	55
3.2.2	Verificação das Sub-rotinas	57
3.2.3	Avaliação do Número de Quadraturas para CFB	60
3.2.4	Algoritmo Produto-Diferença	62
3.2.5	Reconstrução de Função Densidade de Probabilidade	63
3.2.5.1	Máxima entropia	63

3.2.5.2	<i>Spline</i> cúbica	64
3.2.6	Ciclone com Alta Carga	66
3.2.6.1	Teste de independência de malhas	67
3.2.6.2	Análise da influencia do número de pontos de quadratura	69
3.2.6.3	Análise da influencia do arraste sólido-sólido	72
3.2.6.4	Avaliação de outras condições operacionais	72
4	RESULTADOS	73
4.1	IMPLEMENTAÇÃO E VERIFICAÇÃO DAS SUB-ROTINAS	73
4.2	AVALIAÇÃO DO NÚMERO DE QUADRATURAS	77
4.2.1	Reconstrução da Função Densidade de Probabilidade	77
4.2.2	Simulação do Leito Fluidizado Circulante	80
4.3	CICLONE COM ALTA CARGA	85
4.3.1	Teste de Independência de Malhas	85
4.3.2	Análise da Influência do Número de Pontos de Quadratura	86
4.3.3	Análise da Influência do Arraste Sólido-Sólido	95
4.3.4	Avaliação de Outras Condições Operacionais	96
5	CONCLUSÃO	99
	Referências	101
	Apêndices	107
	APÊNDICE A IMPLEMENTAÇÃO DO BALANÇO POPULACIONAL	108
	APÊNDICE B CONDIÇÕES DE CONTORNO	126
	B.1 PERFIL DE VELOCIDADE	126
	B.2 ENTRADA DE SÓLIDOS	128
	B.3 SAÍDA DOS PONTOS DE QUADRATURA	131
	APÊNDICE C ALGORITMO PRODUTO-DIFERENÇA	132
	Anexos	134
	ANEXO A MÉTODOS DE RECONSTRUÇÃO DA PDF	135
	A.1 MÁXIMA ENTROPIA	135
	A.2 <i>SPLINE</i> CÚBICA	137
	ANEXO B SUB-ROTINA PARA CÁLCULO DE AUTOVALORES E	
	AUTOVETORES	151

1 INTRODUÇÃO

Ciclones são dispositivos amplamente utilizados na indústria para separação de sólidos a partir de correntes gasosas devido características de fácil construção, baixo custo de operação e manutenção, alta eficiência de separação para partículas maiores que $5\ \mu m$, além da capacidade de suportar altas temperaturas e pressões. Em especial, ciclones com alta carga de sólidos são utilizados, por exemplo no primeiro estágio de craqueamento catalítico fracionado (*fluid catalytic cracking* - FCC), na indústria de cimento, de mineração, dentre outras. Figura 1 apresenta uma torre de ciclones da indústria de cimento, aplicação típica de ciclones que opera a alta carga e temperatura.

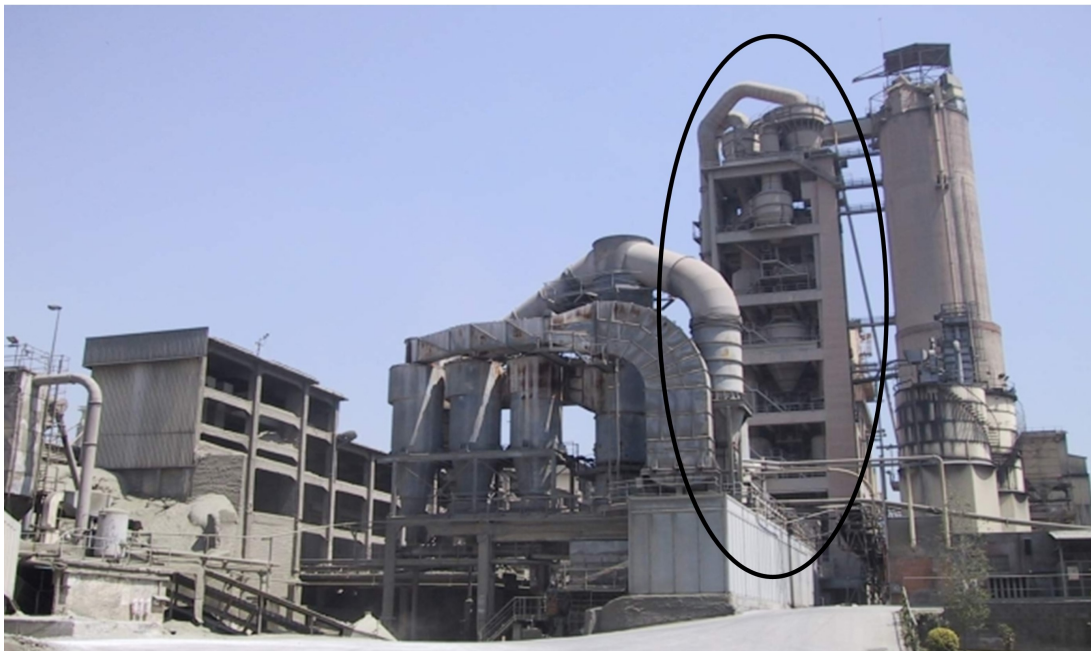


Figura 1 – Ciclones na indústria de produção de cimento (domínio público)

Apesar da simplicidade de sua construção, ciclones apresentam características fenomenológicas complexas podendo conter zonas de reversão do escoamento, regiões de recirculação, alta intensidade de turbulência, dentre outros fenômenos. Deste modo, a técnica de fluidodinâmica computacional (*computational fluid dynamics* - CFD), aliada aos estudos experimentais tradicionalmente conduzidos, apresenta-se como uma alternativa viável para compreensão da fenomenologia do ciclone.

Além disso, os seus principais parâmetros de desempenho, a queda de pressão e a eficiência de coleta, são foco de muitos trabalhos que visam melhorar a eficiência de coleta com menor custo energético possível. Contudo, dados experimentais são restritos a faixas de condições operacionais e geométricas, sendo necessário sempre desenvolver novas

correlações para novos materiais e novas condições. Assim, um modelo fluidodinâmico baseado nas ferramentas de CFD pode contribuir muito com a redução de tempo e custo de projeto e otimização de ciclones.

1.1 MOTIVAÇÃO PELO TEMA

Os primeiros modelos para simulação de ciclones consideravam que o escoamento da fase gasosa não influenciava significativamente o da fase particulada, isto é, resolvia-se o escoamento do gás separadamente e pelo modelo de acompanhamento da trajetória das partículas obtinha-se os resultados para fase sólida. Entretanto, foi observado que as partículas sólidas, mesmo em pequenas quantidades, têm grande influência sobre o escoamento do gás. Deste modo, para ciclones com alta carga, nos quais a concentração de partículas sólidas no escoamento gás-sólido é alta, faz-se necessário considerar a interação entre as fases.

Uma abordagem para a simulação de ciclones de alta carga é a Euleriana-Lagrangeana na qual a fase gasosa é modelada segundo as equações baseadas em um referencial fixo e cada partícula é modelada obedecendo as equações de movimento de Newton. Como esta abordagem analisa o comportamento de cada partícula acarreta em um alto custo computacional, especialmente em condições de alta carga de sólidos. Outra abordagem é a Euleriana-Euleriana, na qual ambas as fases são modeladas segundo um referencial fixo, sendo as partículas consideradas como um pseudo-fluido. Uma desvantagem desta abordagem é o fato de que as partículas são caracterizadas apenas por um diâmetro médio, o que resulta na impossibilidade de avaliação, por exemplo, da curva de eficiência do processo.

Deste modo, deseja-se avaliar se uma abordagem Euleriana-Euleriana acoplada às equações de balanço populacional, utilizadas para caracterizar a distribuição do tamanho de partícula, será capaz de gerar uma curva de eficiência para o processo. Através disso, espera-se obter um modelo que se aproxime das características reais do escoamento gás-sólido polidisperso em ciclone de alta carga e que tenha um custo computacional relativamente baixo no intuito de facilitar o desenvolvimento de novos projetos industriais e auxiliar na otimização de condições operacionais em ciclones de alta carga já em funcionamento.

1.2 PROBLEMA, HIPÓTESES E QUESTIONAMENTOS

A problemática proposta foi a utilização de uma abordagem matemática na qual fosse possível obter resultados para ciclones com alta carga de sólidos considerando um sistema polidisperso em um menor tempo computacional possível. Para tanto, foram

assumidas algumas hipóteses:

- Ambas as fases serão modeladas utilizando uma abordagem Euleriana, de modo que a fase particulada será tratada como um pseudo-fluido caracterizado por um diâmetro médio;
- A polidispersidade dos sólidos será adicionada através das equações de balanço populacional, de modo que o diâmetro médio utilizado na caracterização da fase sólida será atualizado para cada célula computacional em todas as iterações.

Assim, desejou-se responder durante este estudo os seguintes questionamentos:

- É possível representar o comportamento fluidodinâmico de um sistema gás-sólido polidisperso utilizando uma abordagem Euleriana-Euleriana acoplada às equações de balanço populacional?
- É possível calcular a curva de eficiência de coleta de um ciclone de alta carga através da reconstrução da distribuição do tamanho de partículas coletadas a partir dos momentos calculados após a simulação utilizando a abordagem Euleriana-Euleriana acoplada às equações de balanço populacional?

1.3 OBJETIVOS GERAIS

O presente estudo teve como objetivo geral obter uma curva de eficiência para ciclones de alta carga utilizando a abordagem Euleriana-Euleriana acoplada à equação de balanço populacional (*population balance equation* – PBE) que foi resolvida pelo método de quadratura direta baseado em momentos (*direct quadrature method of moments* – DQMOM).

1.3.1 OBJETIVOS ESPECÍFICOS

Objetivos específicos dessa dissertação foram:

- Modelar matematicamente o escoamento gás-sólido em um ciclone de alta carga a luz das PBE's;
- Implementar as funções definidas pelo usuário (*user define function* - UDF) para acoplar a PBE ao código comercial, Fluent;
- Verificar as UDF através de dados da literatura;

- Desenvolver o modelo e gerar a curva de eficiência para o ciclone de alta carga;
- Analisar a curva de eficiência com os dados da literatura.

1.4 ESTRUTURA DA DISSERTAÇÃO

Este estudo foi dividido em cinco capítulos, os quais estão resumidamente descritos abaixo:

- **Capítulo 1:** corresponde a introdução ao tema, bem como sua motivação. Além das hipóteses propostas, perguntas de pesquisa e objetivos deste trabalho;
- **Capítulo 2:** expõe uma breve explanação sobre ciclones citando aspectos geométricos, fenomenológicos e os principais parâmetros de desempenho. Ainda, apresenta uma revisão de literatura relacionada a estudos de ciclones e os pontos de investigação levantados por esta;
- **Capítulo 3:** apresenta a modelagem matemática da abordagem proposta e também trás o que foi necessário implementar para acoplar as equações de balanço populacional ao modelo Euleriano-Euleriano já disponível no *software* comercial Fluent 17.2. Este capítulo ainda apresenta o trabalho de Niemi (2012), que estudou um leito fluidizado pseudo-bidimensional, utilizado para verificar o código implementado. Enfim, descreve a geometria, malha e condições de contorno utilizadas no estudo do ciclone com alta carga de sólidos;
- **Capítulo 4:** apresenta os resultados obtidos para a verificação do código e avaliação do número de quadraturas. Também são apresentados neste capítulo os resultados obtidos para o comportamento fluidodinâmico polidisperso em ciclones com alta carga, bem como as curvas de eficiência obtidas utilizando diferentes números de pontos de quadratura;
- **Capítulo 5:** abrange as principais conclusões resultantes dos resultados obtidos neste estudo, assim como, as sugestões para trabalhos futuros levantadas durante a realização deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica para o entendimento básico de ciclones, bem como suas características geométricas, sua fenomenologia e seus principais parâmetros de desempenho. Também apresenta uma discussão sobre a distribuição do tamanho de partículas e uma revisão bibliográfica.

2.1 CICLONES

Ciclones possuem geometrias simples sem peças móveis, composta basicamente por uma seção cilíndrica sobre uma seção cônica, o que confere suas características de fácil construção, baixo custo de operação e manutenção. Estes equipamentos possuem basicamente oito parâmetros geométricos, como mostrado na Figura 2, na qual D_c representa o diâmetro do corpo do ciclone; D_d , diâmetro da saída de sólidos; D_e , diâmetro da saída de gás; L_c , a altura total do ciclone; L_d , altura da seção cônica; L_e , comprimento do tubo de saída de gás; L_a e L_b representam, respectivamente, altura e largura da entrada de gás-sólido.

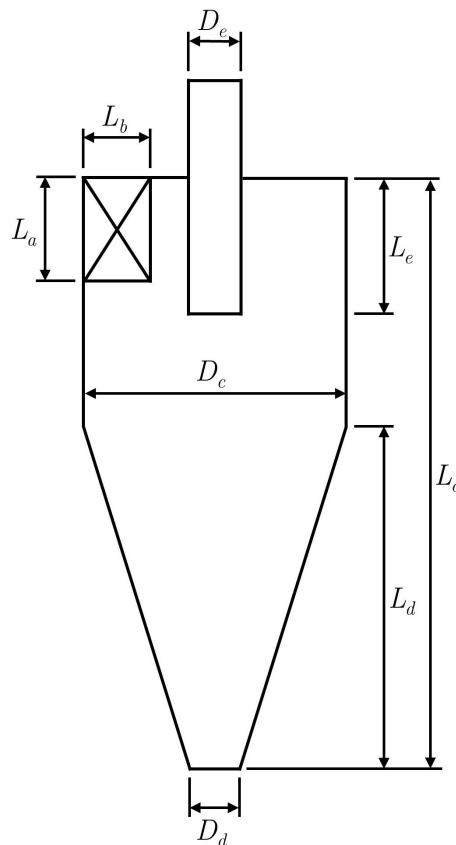


Figura 2 – Esquema da geometria de um ciclone mostrando os oito parâmetros principais

Estes parâmetros geométricos podem assumir diversos valores na busca de se obter uma melhor configuração para uma determinada condição de desempenho. É comum esses parâmetros serem descritos como uma função do diâmetro da parte cilíndrica (D_c) como, por exemplo, os modelos Stairmand (1949) e Lapple (1950), cujos os valores estão descritos na Tabela 1.

Tabela 1 – Parâmetros geométricos dos modelos de ciclones mais conhecidos

Dimensão	Stairmand (1949)	Lapple (1950)
D_d/D_c	0,375	0,25
D_e/D_c	0,5	0,5
L_a/D_c	0,5	0,5
L_b/D_c	0,2	0,25
L_c/D_c	4,0	4,0
L_d/D_c	2,5	2,0
L_e/D_c	0,5	0,625

Além destes parâmetros geométricos, outra possibilidade de otimizar as condições de desempenho de um ciclone é alterar o ângulo e tipo de entrada (Figura 3).

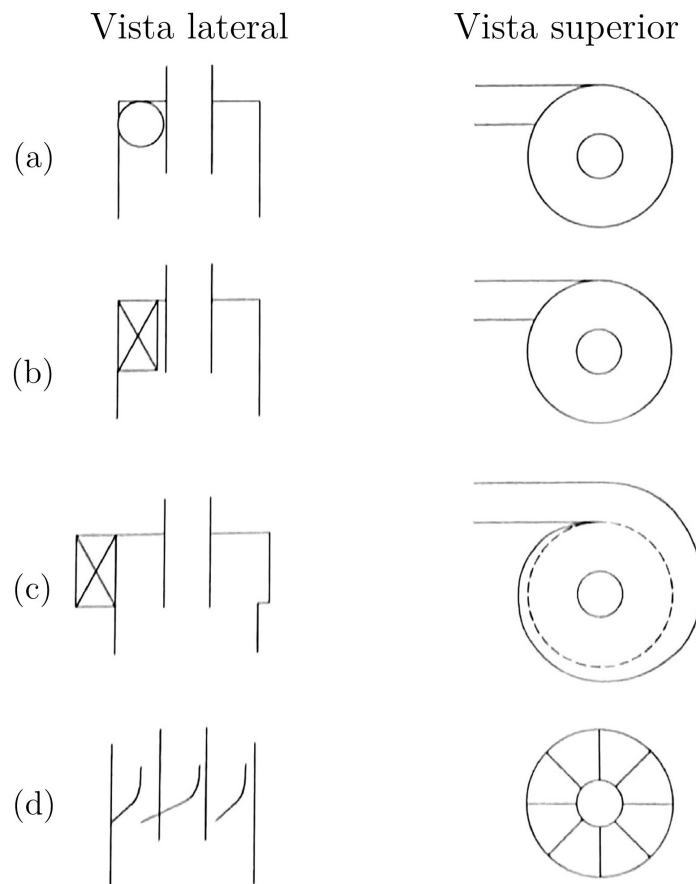


Figura 3 – Tipos de entradas em ciclone a) circular b) tangencial c) voluta d) axial (Adaptado de Hoffmann e Stein (2002))

A Figura 3 mostra quatro possíveis configurações de entradas: circular, tangencial, voluta e axial. A primeira (Figura 3 a) é a mais simples e é normalmente utilizada em ciclones com menor custo de construção e que é aceitável uma perda de eficiência. A segunda (Figura 3 b) é a mais comum na indústria em geral. Essa entrada apresenta uma eficiência de coleta maior que a primeira mas necessita normalmente de uma adaptação dos dutos circulares para a entrada retangular. A voluta (Figura 3 c) é amplamente utilizada para ciclones de alta carga devido ao fato que muitas partículas são separadas antes mesmo de entrarem na parte cilíndrica do ciclone. Diversos ângulos de entrada podem ser usados para este tipo de entrada os mais comuns são 360° (mostrado na Figura 3 c), 270° e 180° . Já a ultima (Figura 3 d) é mais utilizada em separadores com corpo cilíndrico e normalmente são pequenos.

Apesar de possuir uma geometria simples, ciclones possuem um comportamento fenomenológico complexo que será descrito na próxima seção.

2.1.1 Fenomenologia

O comportamento fluidodinâmico de um escoamento gás-sólido é sintetizado na Figura 4, sugerida por Ogawa (1997) e reproduzida por Meier (1998). O escoamento gás-sólido entra tangencialmente ao ciclone e escoam em um movimento giratório. A força centrífuga gerada pelo movimento faz com que as partículas migrem em direção à parede externa do ciclone, enquanto que a força de arraste carrega algumas partículas para o tubo de saída (*vortex finder*). Já o gás começa um escoamento giratório com alta velocidade em direção a parte cônica do ciclone. Este movimento é o mecanismo dominante para coleta de sólidos no *hopper*. Em ciclones verticais, a gravidade vai assistir este movimento mas ela tem uma influência significativa apenas em ciclones de alta carga (HOFFMANN; STEIN, 2002). Na parte cônica do ciclone, há a reversão de escoamento para a fase gasosa sendo que no centro do ciclone, o gás escoam em um movimento giratório em direção ao tubo de saída. Dependendo da geometria e do comportamento fluidodinâmico, o gás pode apresentar regiões de recirculação que elevam o tempo de residência podendo diminuir a eficiência de coleta das partículas e também aumentam a corrosão da parede do ciclone (MEIER, 1998).

Devido à reversão do escoamento, o perfil de velocidade axial apresenta valores positivos próximo ao centro do ciclone e valores negativos próximo à parede considerando as coordenadas assumidas na Figura 4. A velocidade tangencial apresenta um comportamento conhecido como “Vórtice de Rankine”. Este perfil está representado na Figura 5 e é caracterizado por uma zona central com vórtice forçado,

$$v_\theta = \Omega r, \quad (2.1)$$

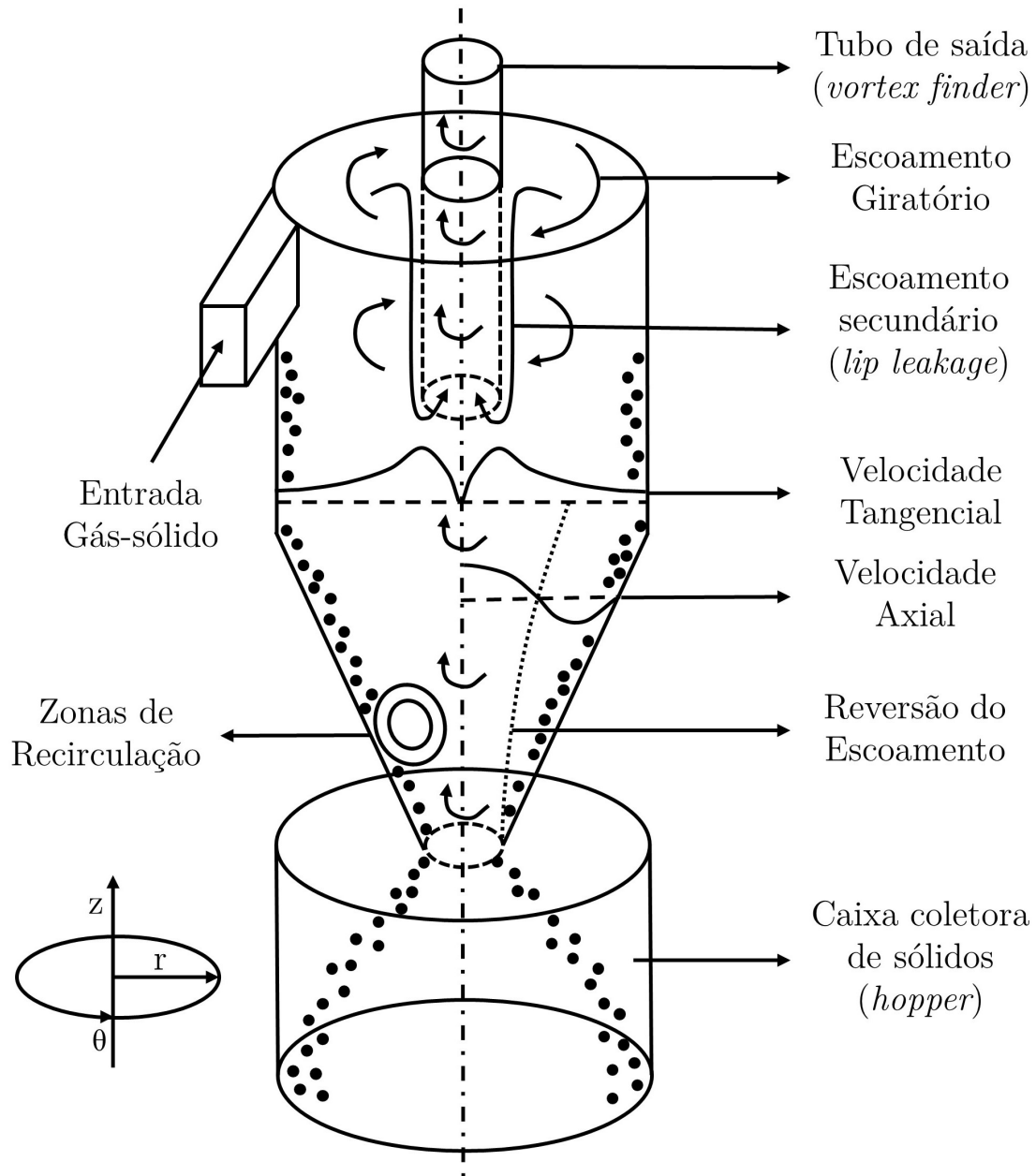


Figura 4 – Características fenomenológicas de um ciclone (Adaptado de Ogawa (1997) *apud* Meier (1998))

na qual Ω é velocidade angular, cercada por uma zona de vórtice livre,

$$v_{\theta} = \frac{C}{r}, \quad (2.2)$$

na qual C é uma constante. Já a velocidade radial não possui um perfil característico, normalmente ela assume valores maiores perto da entrada do *vortex finder* devido ao escoamento secundário (*lip leakage*) que é o escoamento próximo à parede do *vortex finder* (HOFFMANN; STEIN, 2002).

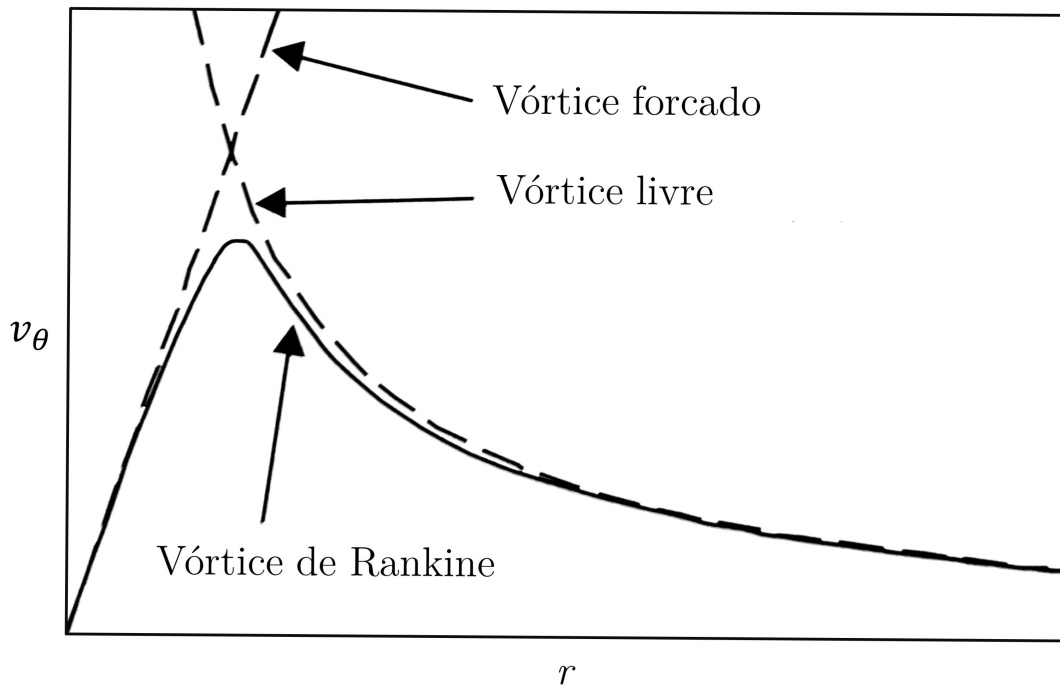


Figura 5 – Perfil da velocidade tangencial (Adaptado de Hoffmann e Stein (2002))

2.1.2 Parâmetros de desempenho

Os dois principais parâmetros de desempenho de ciclones são a queda de pressão e a eficiência de coleta. O primeiro está relacionado com o gasto energético deste processo e o último, com a capacidade do ciclone de separar as partículas da corrente gasosa. Esses parâmetros são influenciados pela configuração geométrica e pelas condições operacionais. Assim, várias correlações empíricas e semi-empíricas foram propostas de modo que fosse viável o projeto e otimização de ciclones.

2.1.2.1 Queda de pressão

O procedimento experimental para medir a queda de pressão em ciclones é medir a diferença entre a pressão estática na parede da entrada e da saída do *vortex finder*. Sendo assim, deve-se realizar o mesmo procedimento na simulação, não podendo apenas obter a média na seção transversal de saída, uma vez que o escoamento em espiral no *vortex finder* causa uma maior pressão estática na parede comparada a da seção transversal média (HOFFMANN; STEIN, 2002).

A queda de pressão em ciclone está diretamente associada com intensidade do escoamento em espiral, sendo proporcional ao quadrado do fluxo volumétrico, de modo que pode-se definir o número de Euler (HOFFMANN; STEIN, 2002), como:

$$Eu = \frac{\Delta p}{\frac{\rho_g v^2}{2}}. \quad (2.3)$$

A partir dessa relação é possível calcular a queda de pressão conhecidos a velocidade na entrada e a massa específica do gás. Assim, equações empíricas foram criadas de modo a calcular o número de Euler baseado nos parâmetros geométricos. Três das equações mais conhecidas estão listadas na Tabela 2, as quais são válidas para ciclones com entradas tangenciais e com baixa carga de sólidos. A relação de Shepherd e Lapple (1940) é mais restrita sendo válida para geometrias próximas a proposta por Lapple (1950) cobrindo a seguinte faixa:

$$\frac{1}{4} \leq \frac{D_e}{D_c} \leq \frac{1}{2}; \quad \frac{1}{4} \leq \frac{L_a}{D_c} \leq \frac{1}{2}; \quad \frac{1}{12} \leq \frac{L_b}{D_c} \leq \frac{1}{4}. \quad (2.4)$$

Tabela 2 – Modelos empíricos para o calculo da queda de pressão

Modelos	Equações
Shepherd e Lapple (1940)	$Eu = \frac{16L_aL_b}{D_e^2}$
First (1949)	$Eu = \frac{24L_aL_b}{D_e^2} \left[\frac{D_c}{L_d(L_c - L_d)} \right]^{\frac{1}{3}}$
Casal e Martinez-Benet (1983)	$Eu = 11,3 \left(\frac{L_aL_b}{D_e^2} \right)^2 + 3,33$

Foram propostos também alguns modelos semi-empíricos para queda de pressão baseados nas perdas na entrada, no corpo do ciclone e no *vortex finder*, como por exemplo, pode-se citar Stairmand (1949) que leva em consideração o efeito da fricção na parede e Barth (1956) que além disto também considera o efeito da carga de sólidos.

2.1.2.2 Eficiência de coleta

As partículas que entram no ciclone podem ser capturadas no final da parte cônica ou perdidas na saída do *vortex finder*. Por definição, eficiência global de coleta (η) é a massa de sólidos capturados dividido pela massa alimentada no sistema:

$$\eta = \frac{m_{capturada}}{m_{alimentada}} \quad (2.5)$$

Considerando o balanço de massa, a Equação (2.5) pode ser reescrita como:

$$\eta = 1 - \frac{m_{perdida}}{m_{alimentada}} \quad (2.6)$$

Em processos industriais, a eficiência global é um parâmetro muito importante para a avaliação de um sistema. Contudo, para sistemas polidispersos, partículas de tamanhos diferentes possuem eficiência de coleta diferentes. Sendo d o diâmetro de uma partícula, existe uma distribuição do tamanho de partículas em massa ou função de densidade de probabilidade mássica ($f(d)$) associada a esse diâmetro na entrada do ciclone,

na saída da parte cônica e na saída do *vortex finder* (HOFFMANN; STEIN, 2002). Assim, a curva de eficiência de coleta ($\eta(d)$) pode ser escrita como:

$$\eta(d) = \eta \frac{f_{capturada}(d)}{f_{alimentada}(d)} \quad (2.7)$$

ou

$$\eta(d) = 1 - (1 - \eta) \frac{f_{perdida}(d)}{f_{alimentada}(d)} \quad (2.8)$$

A Figura 6 apresenta uma típica curva de eficiência de coleta, a qual possui um formato em “S”. Um parâmetro importante desta curva é o diâmetro de corte (d_{50}), o qual é o diâmetro da partícula que tem probabilidade de 50% de ser coletada. Este parâmetro é utilizado por modelos semi-empíricos para calcular a curva de eficiência. Pode-se citar dois métodos distintos para calcular o diâmetro de corte, um a partir do tempo de trajetória, o qual é calculado igualando o tempo de residência da partícula e o necessário para ela atingir a parede do ciclone partindo de uma posição radial especificada, como exemplo pode-se citar o modelo proposto por Lapple (1950) e o modelo denominado de partícula estática que considera o diâmetro de corte o diâmetro da partícula para o qual as forças centrífuga e de arraste se igualam, como em Barth (1956). Além dos modelos semi-empíricos utilizados para calcular o diâmetro de corte, pode-se citar também modelos empíricos como o de Massarani (1997) que leva em consideração a carga de sólidos alimentada no ciclone para o cálculo da eficiência.

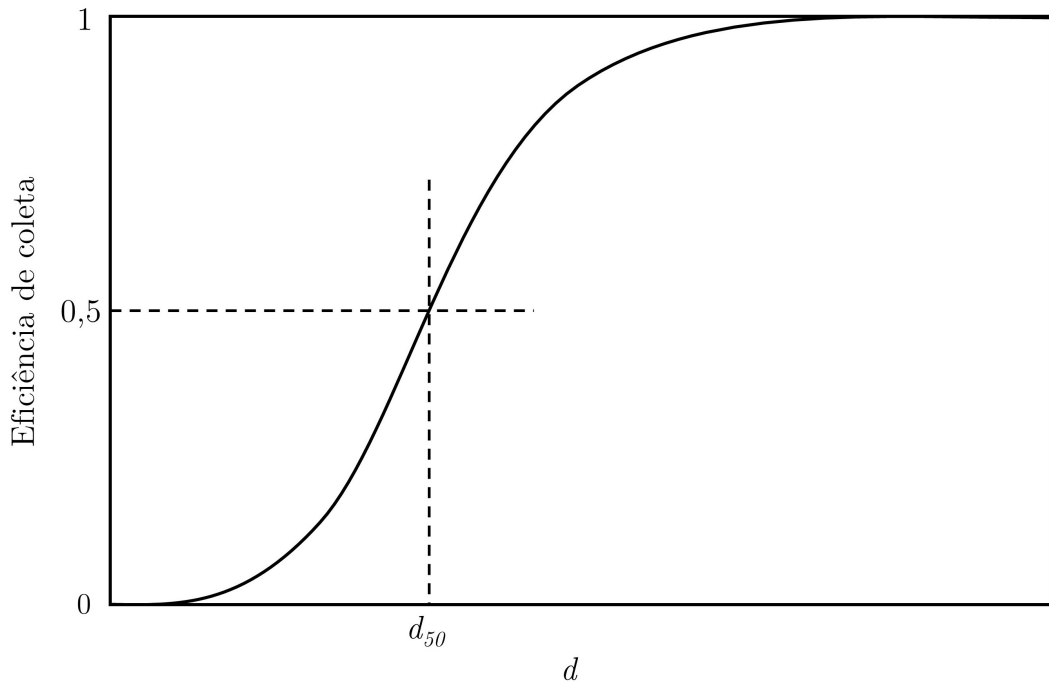


Figura 6 – Curva típica de eficiência de coleta de ciclones

2.2 DISTRIBUIÇÃO DO TAMANHO DE PARTÍCULAS

A distribuição do tamanho de partícula (*particle size distribution* - PSD) pode ser descrita em relação ao número, ao comprimento, à área superficial e ao volume ou massa das partículas (considerando a massa específica das partículas constantes). Sendo o diâmetro uma variável aleatória contínua, pode-se dizer que a distribuição do tamanho de partícula é uma função de densidade. A Figura 7 apresenta a função de densidade de número (*number density function* - NDF) e de volume ou massa.

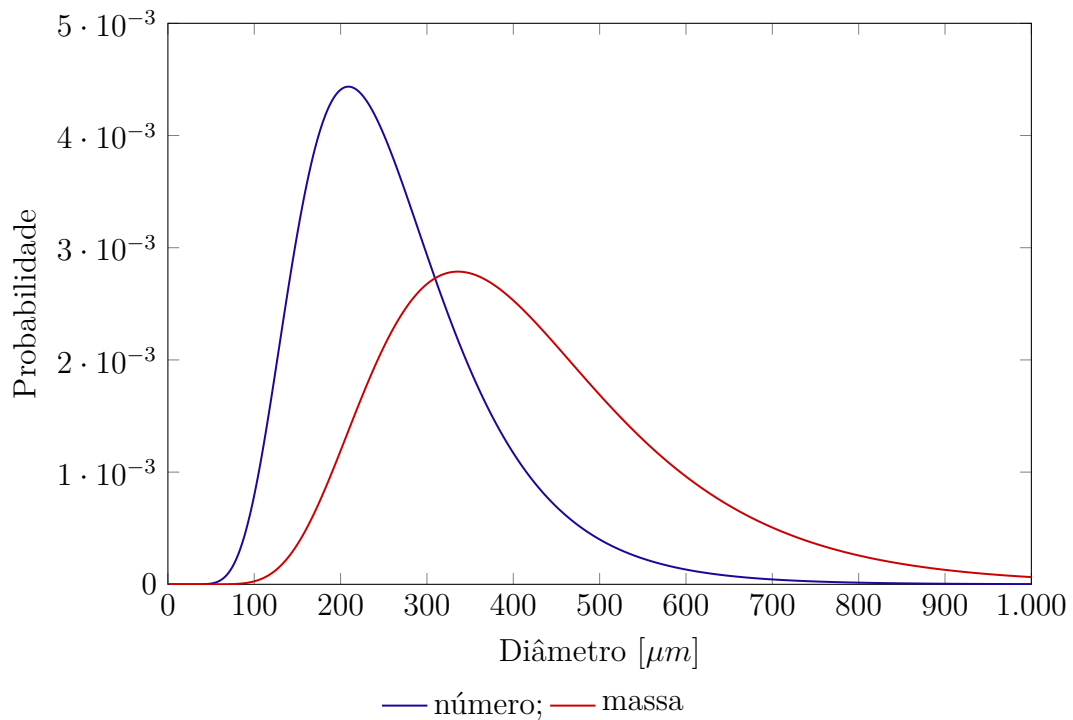


Figura 7 – Exemplo de uma função de densidade de número e de massa para uma determinada amostra de partículas

Como a função densidade de número e de massa representam, respectivamente, o número e a massa de partículas que possuem um determinado diâmetro, era de se esperar que partículas com diâmetros maiores tivessem uma maior representação na distribuição de massa uma vez que o volume destas é proporcional ao cubo do diâmetro. Isso pode ser comprovado na Figura 7 (HOFFMANN; STEIN, 2002).

De modo a caracterizar a função de densidade é muito importante entender o conceito de momento. O k -ésimo momento em torno de zero de uma variável aleatória X é denominado de momento cru ou apenas momento (M) (JAMES, 2010). Este é definido como a esperança de X^k ,

$$M_k = \mathbb{E}(X^k) = \int x^k f(x) dx, \quad (2.9)$$

Pode-se dizer então que para uma NDF, o momento de ordem zero representa o número total de partículas. Caso a função de densidade de número seja normalizada pelo número total de partículas, tem-se uma função de densidade de probabilidade (*probability density function* - PDF), na qual o momento de ordem zero é igual a 1:

$$M_0 = \mathbb{E}(X^0) = \int f(x)dx = 1. \quad (2.10)$$

Para uma PDF, o momento de ordem um representa a média da variável aleatória X e assim, o k -ésimo momento em torno da média é denominado de momento central e é definido como:

$$M_k^C = \mathbb{E}(X - \mathbb{E}(X))^k. \quad (2.11)$$

O segundo momento central é denominado de variância de X e a raiz quadrada da variância é denominada desvio padrão (σ), o qual é a medida de dispersão da PDF. Assim, a variância pode ser escrita em função dos momentos crus como:

$$\begin{aligned} \sigma^2 &= M_2^C = \mathbb{E}(X - \mathbb{E}(X))^2 = \mathbb{E}(X^2 - 2X\mathbb{E}(X) + \mathbb{E}^2) \\ &= \mathbb{E}(X^2) - 2\mathbb{E}(X)\mathbb{E}(X) + \mathbb{E}(X)^2 = \mathbb{E}(X^2) - \mathbb{E}(X)^2 \\ &= M_2 - M_1^2. \end{aligned} \quad (2.12)$$

De modo similar para os momentos centrais de ordem três e quatro tem-se, respectivamente:

$$\begin{aligned} M_3^C &= \mathbb{E}(X - \mathbb{E}(X))^3 = \mathbb{E}(X^3) - 3\mathbb{E}(X^2)\mathbb{E}(X) + 3\mathbb{E}(X)\mathbb{E}(X)^2 - \mathbb{E}(X)^3 \\ &= \mathbb{E}(X^3) - 3\mathbb{E}(X^2)\mathbb{E}(X) + 2\mathbb{E}(X)^3 \\ &= M_3 - 3M_2M_1 + 2M_1^3, \end{aligned} \quad (2.13)$$

$$\begin{aligned} M_4^C &= \mathbb{E}(X - \mathbb{E}(X))^4 = \mathbb{E}(X^4) - 4\mathbb{E}(X^3)\mathbb{E}(X) + 6\mathbb{E}(X^2)\mathbb{E}(X)^2 - 4\mathbb{E}(X)\mathbb{E}(X)^3 + \mathbb{E}(X)^4 \\ &= \mathbb{E}(X^4) - 4\mathbb{E}(X^3)\mathbb{E}(X) + 6\mathbb{E}(X^2)\mathbb{E}(X)^2 - 3\mathbb{E}(X)^4 \\ &= M_4 - 4M_3M_1 + 6M_2M_1^2 - 3M_1^4. \end{aligned} \quad (2.14)$$

Assim, pode-se definir a medida de assimetria de uma PDF, denominada de obliquidade, como:

$$Obliquidade = \frac{M_3^C}{\sigma^3}, \quad (2.15)$$

e a medida de forma de uma função densidade de probabilidade, denominada curtose, como

$$Curtose = \frac{M_4^C}{\sigma^4}, \quad (2.16)$$

Segundo Hoffmann e Stein (2002), pode se ter vantagens em ajustar modelos aos dados experimentais obtidos para a distribuição do tamanho de partículas quando se

está trabalhando com sistemas particulados. Em especial para ciclones, as curvas mais frequentemente ajustadas são a normal, log-normal e Rosin-Rammler, sendo que todas essas possuem dois parâmetros ajustáveis. Nas duas primeiras estes são a média e o desvio padrão da distribuição. Esses três modelos estão descritos na Tabela 3.

Tabela 3 – Modelos mais comuns para ajuste de distribuições do tamanho de partículas

Distribuição	Equação
Normal	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$
Log-Normal	$f(\ln(x)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\overline{\ln(x)})^2}{2\sigma^2}}$
Rosin-Rammier	$f(x) = nkx^{n-1}e^{-kx^n}$

2.3 REVISÃO BIBLIOGRÁFICA

Segundo o site de pesquisa Scopus (2019), a cada ano são publicados cerca de 50 trabalhos sobre simulação de separadores ciclônicos utilizando as técnicas de CFD. Assim se faz necessário, entender o avanço dos estudos dos ciclones ao longo dos anos avaliando os principais trabalhos.

A maioria dos trabalhos apresentados antes de meados da década de 90 consideravam apenas os efeitos da fase contínua sobre a fase descontínua, utilizando assim acoplamento em uma via. Entretanto, muitos trabalhos experimentais confirmavam uma influência significativa da fase particulada na redução da queda de pressão em ciclones. Sendo assim, para escoamentos gás-sólido com alta carga de sólidos se faz necessário o uso de acoplamento em duas vias ou quatro vias. O primeiro considera também a influência da fase gasosa sobre a fase sólida e o outro também considera as interações entre partículas.

Para escoamentos multifásicos gás-sólido duas abordagens são tipicamente aplicadas: a Euleriana-Euleriana e Euleriana-Lagrangeana. Na primeira, a fase sólida é tratada como um pseudo-fluido e também é modelada segundo um referencial fixo. Ambas as fases são tidas como contínuas e interpenetrantes de modo que é possível definir frações volumétricas para cada fase em um volume de controle. Já a outra abordagem trata as partículas discretamente e para cada uma delas se faz necessário resolver as equações de movimento de Newton. Para solucionar as equações de todas as partículas o custo computacional é elevado, por isso é comum agrupar um conjunto de partículas de características semelhantes em uma parcela e, assim, resolver as equações de movimento de Newton para cada parcela.

Deste modo, os trabalhos apresentados nessa revisão foram separados nessas

duas abordagens e relatados em ordem cronológica, apesar de poderem ser divididos em dois grupos: otimização de parâmetros geométricos ou condições operacionais e avanços na modelagem e simulação de ciclones ou ciclones com alta carga de sólidos.

2.3.1 Abordagem Euleriana-Euleriana

Meier (1998) desenvolveu o código CYCLO, no qual utilizou uma abordagem Euleriana-Euleriana multifásica bidimensional e implementou um modelo de força de arraste para representar a interação gás-sólido, sendo um dos primeiros a considerar esta interação. Os resultados obtidos conseguiram prever a atenuação do escoamento vorticial e a redução da queda de pressão, entretanto como foi utilizado apenas o diâmetro médio de Sauter para representar as partículas, não foi possível, com essa modelagem, calcular a curva de eficiência do ciclone. Pensando nisto, ele ainda apresentou o modelo Euleriano-Euleriano-Lagrangeano, no qual, a partir dos campos obtidos pelo modelo anterior, a equação Lagrangeana foi utilizada para definir as trajetória das partículas tornando possível a obtenção da curva de eficiência de coleta.

Qian *et al.* (2007) utilizaram uma abordagem Euleriana-Euleriana com quatro fases sólidas de diâmetros de partículas diferentes de modo a obter uma eficiência de coleta para cada um desses diâmetros. Para isso, foi empregando um arraste de quatro vias de modo a considerar a influência que uma fase sólida tem sobre a outra. Eles utilizaram o modelo *Algebraic Slip Mixture Model*, disponível no código comercial Fluent, o qual resolve as equações de continuidade e de quantidade de movimento para a mistura, ou seja, todas as fases sólidas são consideradas como uma só e depois as velocidades de cada fase são determinadas por relações algébricas. Eles avaliaram a influência da concentração de sólidos observando que com o aumento da carga há um aumento da eficiência de coleta e uma diminuição da perda de carga. No entanto o modelo adotado precisa de melhorias principalmente quando se trata de ciclones operando a altas cargas de sólidos.

Costa *et al.* (2013) propuseram avaliar a influência do número de fases sólidas na eficiência global de coleta e queda de pressão em ciclones. As simulações para uma, três e cinco fases sólidas foram realizadas utilizando os códigos *CYCLO - EE₁*, *CYCLO - EE₃* e *CYCLO - EE₅* os quais foram desenvolvidos a partir do trabalho de Meier (1998). Foi observado que utilizando cinco fases sólidas os resultados obtidos para eficiência global e queda de pressão foram mais próximos ao experimental. Além disso, com cinco fases foi possível estimar a curva de eficiência de coleta. Costa *et al.* (2013) também estudaram a influência da interação sólido-sólido, comparando os resultados obtidos pelas simulações com os obtidos experimentalmente ficou evidente a importância do acoplamento de quatro vias, entretanto percebeu-se ainda a necessidade de uma análise de sensibilidade mais profunda dos parâmetros do modelo utilizado para caracterizar a interação entre as partículas.

Choi *et al.* (2013) estudaram o efeito da posição do *vortex finder* tem sobre a queda de pressão, eficiência de coleta e erosão em ciclones, utilizando um arraste de quatro vias. Eles estudaram nove deslocamentos diferentes, aproximando ou afastando o tubo de saída de gás da entrada do ciclone. Para calcular o diâmetro de corte foram realizadas três simulações para cada caso utilizando diâmetro médio de 0,5, 1,0 e 5,0 μm para a fase sólida. Os resultados mostraram que para a geometria com o *vortex finder* mais afastado da entrada do ciclone obteve-se uma redução da queda de pressão, do diâmetro de corte e da taxa de erosão de 43, 63 e 2%, respectivamente.

Sgrott *et al.* (2015) estudaram uma otimização multi-objetiva de sete parâmetros geométricos de um ciclone com $15 \text{ g} \cdot \text{m}^{-3}$ de carga de sólidos e partículas de diâmetro entre 5 e 15 μm utilizando o algoritmo COMPLEX para minimizar a queda de pressão e maximizar a eficiência de coleta. Todas as simulações foram realizadas no código *CYCLO – EE₅* proposto por Meier (1998), o qual utiliza uma abordagem Euleriana-Euleriana com cinco fases sólidas de diâmetros diferentes, sendo que não foi considerado o arraste sólido-sólido por se tratar de um escoamento com baixa carga de sólidos. O modelo otimizado encontrado obteve uma redução da queda de pressão de 6,3% comparado ao modelo de Stairmand (1949) e um aumento de 11,4% comparado ao de Lapple (1950). Além disso, este modelo obteve um aumento de 3,5% e 9,2% da eficiência de coleta comparado aos modelos de Stairmand (1949) e Lapple (1950), respectivamente. A redução da queda de pressão e o aumento da eficiência de coleta em comparação ao ciclone de Stairmand (1949) podem ser explicadas pela redução do pico da velocidade tangencial e também pelo deslocamento do pico de velocidade tangencial em direção à parede do ciclone.

2.3.2 Abordagem Euleriana-Lagrangeana

Bernardo *et al.* (2006) estudaram a influência do ângulo de entrada em um ciclone de uma indústria de cimento utilizando o código comercial CFX e um arraste de uma via. Eles propuseram três configurações diferentes, todas com inclinação positiva de 30°, 45° e 60°. Os resultados demonstraram que as configurações proporcionaram uma menor queda de pressão e uma maior eficiência de coleta, de modo que as angulações podem aumentar a eficiência do equipamento e diminuir o consumo de energia da planta industrial. O trabalho deixa em aberto ainda a possibilidade de otimização do ângulo de entrada.

Derksen *et al.* (2008) avaliaram os efeitos da concentração de sólidos considerando um acoplamento em duas vias entre uma parcela de partículas e a fase gasosa. Eles observaram que com o aumento da concentração houve a diminuição da velocidade tangencial e das flutuações turbulentas indo de acordo com o que já se encontrava na literatura. Eles ainda compararam esse estudo com o Derksen (2003) e observaram que

o acoplamento em duas vias obteve uma eficiência total maior do que quando não se considerava a influência do sólido no escoamento gasoso.

El-Batsh (2013), a partir do modelo padrão de Stairmand (1949), propôs estudar o efeito da variação do diâmetro do *vortex finder* na queda de pressão e na eficiência de coleta. Ele realizou este estudo no código comercial Fluent, utilizando acoplamento em uma via. Foram propostas três configurações para o diâmetro do *vortex finder*: 0,4, 0,5 e 0,6 vezes o diâmetro do ciclone. A partir dos resultados, observou-se que com o aumento do diâmetro houve uma diminuição na queda de pressão, no entanto houve também uma diminuição da eficiência de coleta e assim, conclui que a escolha do melhor diâmetro depende do peso desses dois parâmetros de desempenho para cada caso.

Souza *et al.* (2015) estudaram a influência que o comprimento e a forma do duto de saída de gás tem sobre o perfil fluidodinâmico em simulações de ciclones. Normalmente, nas simulações, o comprimento do *vortex finder* é curto e escolhido arbitrariamente. No entanto, em aparatos ambientais esses dutos costumam ser grandes e conter curvas ao longo do seu comprimento. Assim, eles se propuseram a testar diferentes tamanhos de dutos de saída, bem como, ângulos e raios diferentes para as curvas propostas para o *vortex finder*. As simulações foram realizadas utilizando um arraste de uma via. Os resultados obtidos mostram a influência do comprimento e forma do *vortex finder* no comportamento fluidodinâmico do ciclone. Uma das observações foi que com o aumento do raio da curva no duto de saída existe um aumento da queda de pressão. Outra observação foi que com o aumento do comprimento houve primeiramente uma diminuição da queda de pressão e depois começou a aumentar, havendo um ponto de mínimo para queda de pressão. Entretanto, a influência observada por Souza *et al.* (2015) foi muito menor que a relatada por Schmidt *et al.* (2004). Além disso, a relação encontrada entre tamanho e forma do duto de saída e o diâmetro de corte foi muito complexa, de modo que, mostrou-se necessário, ainda, mais estudos para observar o efeito que a condição de contorno de saída de gás tem sobre o comportamento fluidodinâmico do ciclone.

Brar *et al.* (2015) propuseram avaliar o efeito da variação do comprimento da seção cilíndrica e cônica na eficiência de coleta e perda de carga de um ciclone Stairmand (1949). Foram propostos dez casos distintos com cinco diferentes tamanhos de cada comprimento. Todos os casos foram simulados utilizando o código comercial Fluent. O aumento de ambos os comprimentos resultaram numa diminuição da queda de pressão e no aumento da eficiência de coleta, bem como, a diminuição do diâmetro de corte. Além disso, Brar *et al.* (2015) constataram que a otimização desses parâmetros geométricos é dependente da velocidade de entrada do ciclone. Do mesmo modo, Demir *et al.* (2016) avaliou nove casos com a combinação de três comprimentos para cada parâmetro. A partir das simulações monofásicas destes casos, os autores chegaram na mesma conclusão que o aumento das duas seções do ciclone acarretam na diminuição da perda de carga. Além

disso, eles observaram também que este aumento dos parâmetros geométricos avaliados resultou em um menor pico da velocidade tangencial, e esse resultado já era esperado uma vez que o pico da velocidade tangencial é responsável por praticamente 80% da queda de pressão em ciclones como afirmado por Noriler *et al.* (2004).

Balestrin *et al.* (2017) investigaram a mudança no escoamento gás-sólido gerado pela redução da área transversal ao final do *vortex finder* utilizando um acoplamento em uma via. De acordo com os resultados obtidos através da simulação no código comercial Fluent, foi possível observar que essa redução provocou um escoamento secundário, o qual resultou na mudança do perfil da velocidade tangencial e no aumento da eficiência de coleta de partículas com diâmetro menor que $5\ \mu\text{m}$, mostrando-se uma alternativa viável para algumas aplicações de ciclone.

Huang *et al.* (2017) propuseram a adição de 15 tubos com $6\ \text{mm}$ de diâmetro e $20\ \text{mm}$ de comprimento na entrada tangencial do ciclone dispostos em cinco fileiras de três tubos de modo a estudar a influência destes no comportamento fluidodinâmico e nos parâmetros de desempenho do ciclone. A Figura 8 apresenta um esquema da adição destes tubos. Todas as simulações realizadas por Huang *et al.* (2017) foram realizadas no *software* comercial Fluent 17.0 utilizando um acoplamento de uma via. Eles observaram que, para uma velocidade de entrada de $11\ \text{m} \cdot \text{s}^{-1}$, a adição dos tubos resultou num aumento de 8,1% na queda de pressão, redução de aproximadamente 2% do pico da velocidade tangencial e na diminuição do diâmetro de corte de $2,04\ \mu\text{m}$ para $1,89\ \mu\text{m}$. De modo similar, Fatahian *et al.* (2018) propôs a adição destes 15 tubos na entrada do ciclone e também a adição de 7 tubos de $5,25\ \text{mm}$ de diâmetro e $20\ \text{mm}$ de comprimento na saída do *vortex finder*. Os resultados encontrados foram similares ao de Huang *et al.* (2017), de modo que os tubos acrescentados acarretaram no aumento da perda de carga e da eficiência de coleta.

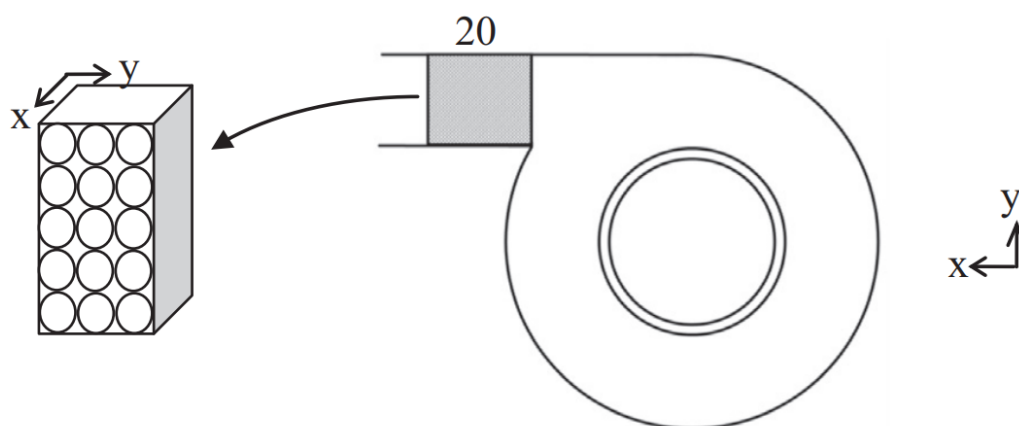


Figura 8 – Esquema da adição de 15 tubos com $6\ \text{mm}$ de diâmetro e $20\ \text{mm}$ de comprimento na entrada tangencial do ciclone (Adaptado de Huang *et al.* (2017))

Wasilewski e Brar (2017) estudaram o efeito de diferentes ângulos da voluta e diâmetros do *vortex finder* na curva de eficiência de coleta e na queda de pressão. Eles propuseram três variações diferentes esquematizadas na Figura 9. A primeira e a segunda variação apresentam o mesmo diâmetro do duto de saída e um ângulo da voluta de 180° e 270° , respectivamente. Já a terceira possui um maior diâmetro do *vortex finder* e um ângulo de 180° . Todas as simulações foram realizadas no código comercial Fluent utilizando um acoplamento em uma via. Baseados nos resultados obtidos, Wasilewski e Brar (2017) constataram que com o aumento do ângulo da voluta e a diminuição do diâmetro de entrada do *vortex finder* obtinha-se uma maior eficiência de coleta e uma maior queda de pressão.

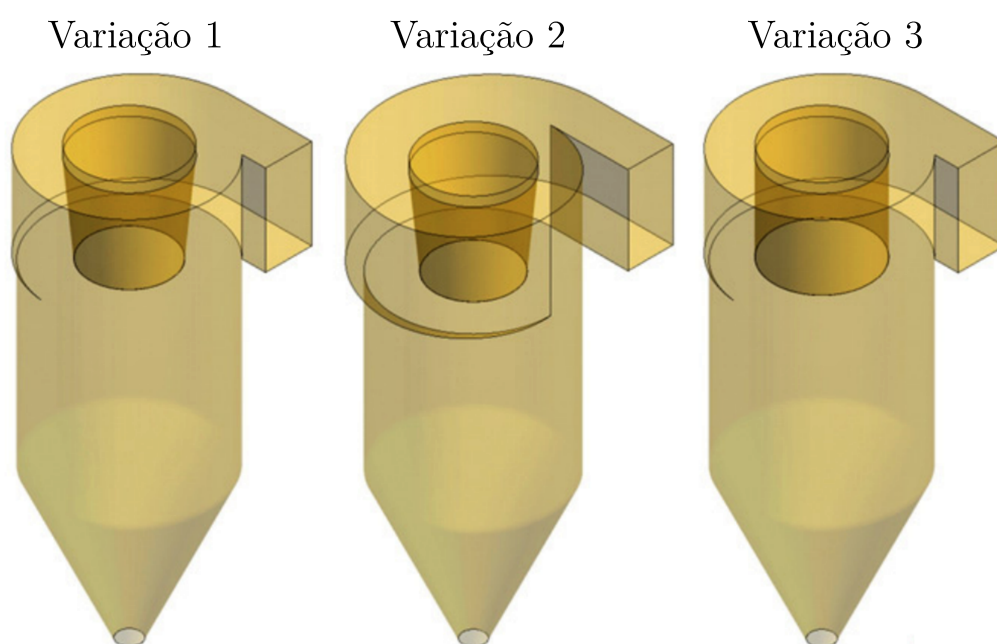


Figura 9 – Esquema das três variações propostas por Wasilewski e Brar (2017) (Adaptado de Wasilewski e Brar (2017))

Sun *et al.* (2017) propuseram uma otimização multi-objetivo de dez parâmetros geométricos do ciclone utilizando a metodologia de superfície de resposta e as técnicas de fluidodinâmica computacional para minimizar a queda de pressão e maximizar a eficiência de coleta. As simulações foram realizadas no código comercial Fluent 16.1 utilizando um acoplamento de uma via. Foi obtido pela metodologia de superfície de resposta um modelo ótimo que reduzia a queda de pressão em 20,7% e o diâmetro de corte em 75,38% comparado ao ciclone de Stairmand (1949). Ao encontro do resultado obtido por Sgrott *et al.* (2015), novamente, pode-se explicar o melhor desempenho do ciclone otimizado pela redução do pico da velocidade tangencial e do deslocamento deste pico em direção à parede do ciclone.

Kozolub *et al.* (2017) utilizaram uma abordagem Euleriana-Lagrangeana e um

acoplamento em quatro vias para estudar o escoamento com alta carga de sólidos em ciclone. Eles observaram que com aumento da carga de sólidos houve uma atenuação do pico da velocidade tangencial e um aumento da queda de pressão. No entanto, o estudo subestimou a queda de pressão. Os autores sugerem realizar uma análise de sensibilidade dos parâmetros dos modelos utilizados na simulação numérica a fim de melhorar a previsão da queda de pressão.

Ashry *et al.* (2018) estudaram o efeito da esfericidade da partícula na queda de pressão de ciclones. Eles realizaram estudos experimentais e numéricos para três compostos: cimento, ureia e poliestireno, em ordem crescente de esfericidade. As simulações numéricas foram realizadas no Fluent utilizando um acoplamento de uma via e variando o coeficiente de arraste. Foram utilizados o coeficiente descrito por Kasper (1982) para partículas esféricas e os coeficientes de Haider e Levenspiel (1989), Chien *et al.* (1994) e Swamee e Ojha (1991) para partículas não esféricas. Para o poliestireno cuja a esfericidade é próxima de 1, o modelo de partícula esférica foi o que mais se aproximou do resultado experimental, em contraposição, para o cimento que possui a menor esfericidade entre os três compostos, o modelo de Haider e Levenspiel (1989) obteve resultados melhores. Em adição, foi observado também que o aumento da esfericidade acarreta no aumento da queda de pressão em ciclones.

Zhou *et al.* (2018) investigaram o efeito da rugosidade da parede no comportamento fluidodinâmico e no desempenho de um ciclone Stairmand (1949). Os autores propuseram avaliar a rugosidade na faixa de 0,01 a 2 mm utilizando uma carga de sólidos de $20 \text{ g} \cdot \text{m}^{-3}$ e velocidade de entrada variando de 10 a $25 \text{ m} \cdot \text{s}^{-1}$. Todas as simulações foram realizadas no Fluent utilizando um acoplamento de duas vias. Foi observado que para velocidades de entrada baixas a rugosidade quase não tinha influência sobre a queda de pressão, contudo, para altas velocidades o aumento da rugosidade diminuía a queda de pressão. Em relação a eficiência de coleta, existe um ponto de máxima eficiência próxima à rugosidade de 0,1 mm. Além disso, o aumento da rugosidade tende a diminuir o pico de velocidade tangencial e aumentar a velocidade axial em ciclones.

Huang *et al.* (2018) estudaram o efeito do aumento da carga de sólidos na eficiência de coleta. Todas as simulações foram realizadas no *software* comercial Fluent 17.1 variando a carga alimentada no ciclone entre 1,6 e $115,3 \text{ g} \cdot \text{m}^{-3}$ e utilizando um acoplamento em duas vias. Foi observado experimentalmente que a eficiência de coleta aumenta com o aumento da carga de sólidos, no entanto através das simulações quase não foi possível observar diferenças entre as curvas de eficiência de coleta, sendo as curvas muito próximas ou até mesmo mostrando uma relação oposta à obtida experimentalmente. Além disso, comparando a curva de eficiência de coleta obtida numericamente com a experimentalmente (Figura 10), observa-se que a eficiência foi superestimada para diâmetros menores que $1,5 \mu\text{m}$.

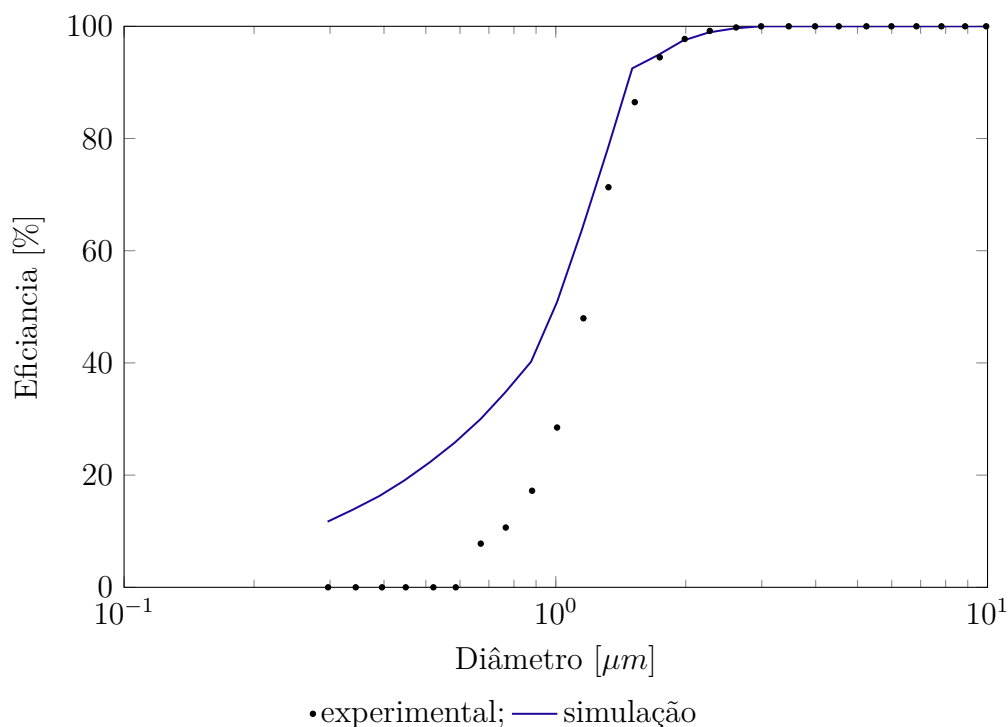


Figura 10 – Comparação da curva de eficiência experimental e numérica obtida por Huang *et al.* (2018) para o ciclone com carga de sólidos de $115,3 \text{ g/m}^3$

2.4 PONTO DE INVESTIGAÇÃO

A Tabela 4 apresenta a síntese dos trabalhos avaliados na revisão bibliográfica, analisando estes quanto à abordagem utilizada Euleriana-Euleriana (E-E) ou Euleriana-Lagrangeana (E-L) e quanto ao arraste utilizado avaliando de a interação foi de uma, duas ou quatro vias. Além de avaliar objetivo dos trabalhos dividindo-os em otimização de parâmetros geométricos ou operacionais e na proposição de novos modelos para simulação de ciclones.

Observa-se pela Tabela 4 que a maioria dos trabalhos realizados atualmente utilizam abordagem Euleriana-Lagrangeana, uma vez que com esta obtém-se diretamente a curva de eficiência de coleta. No entanto, esta abordagem demanda um custo computacional elevado, sendo possível observar que os trabalhos atuais, mesmo para baixa carga de sólidos, utilizam um modelo de arraste de uma via ou duas vias. Assim, esta metodologia mostra-se inviável em relação ao tempo computacional quando se deseja simular ciclones operando a alta carga de sólidos utilizando um acoplamento de quatro vias.

Uma alternativa de tentar reproduzir a curva de eficiência de coleta observada na literatura é a utilização de uma abordagem Euleriana-Euleriana com mais de uma fase particulada. Alguns questionamentos podem ser feitos sobre a utilização dessa abordagem como, por exemplo, definir os diâmetros que cada fase e quantas fases são necessárias para construir a curva de eficiência. Sendo que, dependendo do número de fases sólidas

Tabela 4 – Síntese dos trabalhos avaliados na revisão bibliográfica

Estudos	Abordagens		Arraste (vias)			Objetivo	
	E-E	E-L	1	2	4	Otimização	Modelo
Meier (1998)	x			x			x
Bernardo <i>et al.</i> (2006)		x	x			x	
Qian <i>et al.</i> (2007)	x				x		x
Derksen <i>et al.</i> (2008)		x		x			x
Costa <i>et al.</i> (2013)	x			x	x		x
Choi <i>et al.</i> (2013)	x				x	x	
El-Batsh (2013)		x	x			x	
Sgrott <i>et al.</i> (2015)	x			x		x	
Souza <i>et al.</i> (2015)		x	x			x	
Brar <i>et al.</i> (2015)		x	x			x	
Demir <i>et al.</i> (2016)		x	x			x	
Balestrin <i>et al.</i> (2017)		x	x			x	
Huang <i>et al.</i> (2017)		x	x			x	
Wasilewski e Brar (2017)		x	x			x	
Sun <i>et al.</i> (2017)		x	x			x	
Kozolub <i>et al.</i> (2017)		x			x		x
Fatahian <i>et al.</i> (2018)		x	x			x	
Ashry <i>et al.</i> (2018)		x	x				x
Zhou <i>et al.</i> (2018)		x		x			x
Huang <i>et al.</i> (2018)		x		x			x

necessárias, o custo computacional desta metodologia pode também ser elevado.

Além disso, as modelagens de ciclones com alta carga de sólidos apresentaram resultados não muito próximos aos experimentais como mostrado na Figura 10. Com isso, pretende-se neste trabalho utilizar a abordagem Euleriana-Euleriana e para determinar a curva de eficiência de coleta será acoplado à esta abordagem as equações de balanço populacional a luz de trabalhos já realizados para outros sistemas, como os de Mazzei (2008) que utilizou a PBE para estudar a fluidização de dois compostos com distribuição de tamanhos de partículas diferentes; Dutta *et al.* (2012) que empregaram a PBE para caracterizar os diferentes tamanhos das partículas de catalizador em um *riser* de FCC; Niemi (2012) usou a PBE junto com uma abordagem de mistura para avaliar o escoamento em um leito fluidizado circulante (*circulating fluidized bed* - CFB) e Yuan *et al.* (2014) que utilizou a PBE para avaliar o comportamento do escoamento de tamanhos diferentes de bolhas em uma coluna de bolhas.

3 METODOLOGIA

A metodologia foi separada em duas partes: modelagem e simulação. A primeira é composta pela formulação matemática e a segunda apresenta as características numéricas dos casos de estudo e metodologia de análise de dados.

3.1 MODELAGEM MATEMÁTICA

Com o propósito desenvolver a modelagem do ciclone operando a alta carga de sólidos, foi proposto um modelo Euleriano-Euleriano e suas equações de fechamento acoplado às equações de balanço populacional com suas hipóteses simplificadoras. Além da abordagem de mistura utilizada para calcular as velocidades dos pontos de quadratura necessárias para a solução do balanço populacional.

3.1.1 Desenvolvimento do Modelo Euleriano-Euleriano

O modelo proposto foi baseado no trabalho de Niemi (2012), no qual foram assumidas as hipóteses do contínuo e da interpenetrabilidade entre as fases. A primeira considera que todas as fases são consideradas contínuas, ou seja, as moléculas ou partículas são pequenas o suficiente para que não haja espaços vazios entre elas, que as suas propriedades sejam uniformes e que o fluido seja grande suficiente para conter um número muito grande de moléculas ou partículas. A última complementa a hipótese do contínuo e permite que duas fases ocupem o mesmo volume no espaço ao mesmo tempo, sendo possível definir a fração volumétrica (α) de cada fase (MEIER, 1998). Além disso, foram assumidas as seguintes hipóteses:

- a fase continua em escoamento turbulento pode ser representada via equações de médias de Reynolds (*Reynolds Averaged Navier-Stokes* - RANS), nas quais uma variável “x” qualquer pode ser decomposta em sua média e sua flutuação,

$$x = \bar{x} + x'. \quad (3.1)$$

- a turbulência da fase sólida foi desconsiderada bem como sua influência sobre a fase gasosa;
- Após a rerepresentação do escoamento turbulento via decomposição de Reynolds vários fluxos adicionais de massa e quantidade de movimento aparecem

devido às flutuações, caracterizados por correlações de primeira, segunda e terceira ordem. Segundo Meier (1998), a maioria desses termos podem ser desconsiderados, incluindo o termo de dispersão turbulenta. De modo que o termo de maior influência sobre o escoamento turbulento da fase gasosa associado as flutuações é o tensor de Reynolds, o qual é escrito como:

$$\boldsymbol{\tau}_{Rg} = \alpha_g \rho_g \overline{\mathbf{v}'_g \mathbf{v}'_g}, \quad (3.2)$$

na qual α é a fração volumétrica, ρ , a massa específica e \mathbf{v} , o vetor velocidade e o subíndice g representa a fase gasosa;

- A fase sólida foi considerada invíscida;
- As velocidades dos ponto de quadratura foi modelada a partir da abordagem de mistura proposta por Manninen *et al.* (1996).

Deste modo, suprimindo por conveniência os símbolos de média, pode-se escrever a equação da continuidade para o gás:

$$\frac{\partial(\alpha_g \rho_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{v}_g) = 0. \quad (3.3)$$

Do mesmo modo, pode-se escrever a equação da continuidade para a fase particulada:

$$\frac{\partial(\alpha_s \rho_s)}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s) = 0, \quad (3.4)$$

na qual o subíndice s representa a fase sólida.

Para a fase gasosa, a equações de média de quantidade de movimento na sua forma diferencial conservativa pode ser escrita como:

$$\frac{\partial(\alpha_g \rho_g \mathbf{v}_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{v}_g \mathbf{v}_g) = -\alpha_g \nabla p + \nabla \cdot (\boldsymbol{\tau}_g - \boldsymbol{\tau}_{Rg}) + \alpha_g \rho_g \mathbf{g} + K_{gs}(\mathbf{v}_s - \mathbf{v}_g), \quad (3.5)$$

na qual p é a pressão da fase gás, \mathbf{g} , a aceleração da gravidade, K_{gs} , o coeficiente de arrasto entre as fases gasosa e sólida e $\boldsymbol{\tau}_g$, o tensor tensão, assumido como:

$$\boldsymbol{\tau}_g = \alpha_g \mu_g (\nabla \mathbf{v}_g + \nabla \mathbf{v}_g^T) + \alpha_g (\lambda_g - \frac{2}{3} \mu_g) (\nabla \cdot \mathbf{v}_g) \mathbf{I}, \quad (3.6)$$

na qual μ é a viscosidade dinâmica (*shear viscosity*), λ , a viscosidade volumétrica (*bulk viscosity*) e \mathbf{I} , o tensor identidade.

Já para a fase sólida pode-se escrever a equação de conservação da quantidade de movimento como:

$$\frac{\partial(\alpha_s \rho_s \mathbf{v}_s)}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s \mathbf{v}_s) = -\alpha_s \nabla p + \alpha_s \rho_s \mathbf{g} + K_{gs}(\mathbf{v}_g - \mathbf{v}_s). \quad (3.7)$$

Após descritas as equações de continuidade e da quantidade de movimento para ambas as fases, percebe-se a necessidade de modelos adicionais para o coeficiente de arraste, o diâmetro característico e o tensor de Reynolds.

3.1.2 Modelo de Coeficiente de Arraste

O modelo de coeficiente de arraste utilizado nesse trabalho foi a abordagem proposta por Gidaspow (1994), o qual é uma combinação dos modelos de Ergun (1952) e Wen e Yu (1966). O modelo de Ergun (1952) é considerado válido para escoamentos densos e é dado por:

$$K_{gs} = K_E = 150 \frac{\alpha_s(1 - \alpha_g)\mu_g}{\alpha_g d_s^2} + 1,75 \frac{\rho_g \alpha_s |\mathbf{v}_g - \mathbf{v}_s|}{d_s}, \quad (3.8)$$

na qual d_s é o diâmetro característico da fase sólida.

Em contra partida o modelo Wen e Yu (1966), considerado mais correto para escoamentos diluídos, é expresso por:

$$K_{gs} = K_{WY} = \frac{3}{4} C_D \frac{\alpha_s \alpha_g \rho_g |\mathbf{v}_g - \mathbf{v}_s|}{d_s} \alpha_g^{-2,65}, \quad (3.9)$$

no qual o coeficiente C_D é dado pela equação:

$$C_D = \begin{cases} \frac{24}{Re_s} [1 + 0,15(Re_s)^{0,687}], & \text{se } Re_s \leq 1000, \\ 0,44, & \text{se } Re_s > 1000, \end{cases} \quad (3.10)$$

na qual o número de Reynolds (Re_s) pode ser escrito como:

$$Re_s = \frac{\alpha_g \rho_g |\mathbf{v}_g - \mathbf{v}_s| d_s}{\mu_g}. \quad (3.11)$$

Deste modo, Gidaspow (1994) combinou estes dois modelos da seguinte maneira:

$$K_{gs} = \begin{cases} K_{WY}, & \text{se } \alpha_s < 0,2, \\ K_E, & \text{se } \alpha_s \geq 0,2. \end{cases} \quad (3.12)$$

3.1.3 Modelo de Diâmetro Característico

O diâmetro característico da fase sólida foi modelado utilizando o diâmetro médio de Sauter (*Sauter mean diameter* - SMD) que, por definição, é a divisão o momento de terceira ordem pelo de segunda ordem da função densidade de probabilidade de número,

$$SMD = \frac{M_3}{M_2}. \quad (3.13)$$

Deste modo, o diâmetro característico para cada célula computacional foi calculado a partir dos pesos e abscisas das quadraturas oriundos da equação do balanço populacional resolvida pelo método de quadratura direta baseado em momentos (DQ-MOM).

3.1.4 Modelo de Turbulência

A partir do estudo pioneiro de Boysan *et al.* (1982), que utilizou uma combinação do modelo $k - \epsilon$ padrão proposto por Launder e Spalding (1974) e os tensores algébricos de Rotta (1951) e Rodi (1982) como modelo de turbulência, vários outros autores vem estudando modificações no modelo de turbulência no intuito de melhorar a representação do perfil da velocidade tangencial.

É, portanto, sabido desde então que uma boa aproximação deste escoamento exige a aplicação de um modelo de turbulência que considera uma característica anisotrópica do sistema. Os modelos anisotrópicos mais utilizado em simulação de ciclones atualmente são os modelos das tensões de Reynolds (*Reynolds Stress Model* - RSM), em especial o de Speziale, Sarkar e Gatski (1991) (RSM-SSG) que utiliza uma modelagem quadrática para o termo de deformação devido a pressão, como pode ser observado nos trabalhos de Balestrin *et al.* (2017), Sedrez *et al.* (2017), Mariani *et al.* (2017) e Wasilewski e Brar (2017). Com isso, este estudo também utilizou o RSM-SSG que adiciona equações de transporte para cada um dos termos do tensor de Reynolds,

$$\frac{\partial(\alpha_g \rho_g \overline{\mathbf{v}'_g \mathbf{v}'_g})}{\partial t} + \nabla \cdot \alpha_g \mathbf{v}_g (\rho_g \overline{\mathbf{v}'_g \mathbf{v}'_g}) = \Psi + \mathbf{D}_M + \mathbf{D}_T + \Pi - \epsilon, \quad (3.14)$$

na qual Ψ é o termo que representa a taxa de produção de tensão turbulenta e é definido por,

$$\Psi = -\rho_g [\overline{\mathbf{v}'_g \mathbf{v}'_g} (\nabla \mathbf{v}_g)^T + (\nabla \mathbf{v}_g) \overline{\mathbf{v}'_g \mathbf{v}'_g}], \quad (3.15)$$

\mathbf{D}_M é o termo da difusão molecular que é escrito como:

$$\mathbf{D}_M = \nabla \cdot (\mu_g \nabla \overline{\mathbf{v}'_g \mathbf{v}'_g}), \quad (3.16)$$

os termos \mathbf{D}_T , Π e ϵ representam, respectivamente, a difusão turbulenta, a deformação devida a pressão e a dissipação. Todos esses termos necessitam ser modelados de modo a realizar o fechamento da Equação 3.14. A difusão turbulenta foi modelada por Lien e Leschziner (1994) como sendo:

$$\mathbf{D}_T = \nabla \cdot \left(\frac{\mu_{t,g}}{\sigma_k} \nabla \overline{\mathbf{v}'_g \mathbf{v}'_g} \right), \quad (3.17)$$

na qual a contante σ_k assume valor de 0,82 e $\mu_{t,g}$ é a viscosidade turbulenta que é definida como:

$$\mu_{t,g} = \rho_g C_\mu \frac{\varepsilon_g^2}{k_g}, \quad (3.18)$$

na qual a constante C_μ assumi valor de 0,09, k_g é a energia cinética turbulenta escrita como:

$$k_g = \frac{1}{2} \overline{\mathbf{v}'_g \mathbf{v}'_g}, \quad (3.19)$$

e ε_g representa a taxa de dissipação da energia cinética turbulenta que por sua vez exige uma equação de transporte adicional:

$$\frac{\partial(\alpha_g \rho_g \varepsilon_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{v}_g \varepsilon_g) = \nabla \cdot \left[\left(\mu_g + \frac{\mu_{t,g}}{\sigma_\varepsilon} \right) \nabla \varepsilon_g \right] + \frac{1}{2} \frac{\varepsilon_g}{k_g} \alpha_g C_{\varepsilon 1} \Psi - \frac{\varepsilon_g^2}{k_g} \alpha_g \rho_g C_{\varepsilon 2}, \quad (3.20)$$

na qual σ_ε assumi o valor de 1,0, $C_{\varepsilon 1}$, o valor de 1,44 e $C_{\varepsilon 2}$, 1,92.

O termo de deformação devido a pressão pode ser modelado de diferentes formas, no entanto, a modelagem quadrática proposta por Speziale *et al.* (1991) vem demonstrando ter um melhor desempenho para representar escoamentos em espiral. Deste modo, neste estudo foi utilizado a modelagem quadrática.

Por fim, o termo de dissipação é modelado como:

$$\epsilon = \frac{2}{3} \delta \rho_g \varepsilon_g. \quad (3.21)$$

3.1.5 Equações de Balanço Populacional

As PBE's vem sendo utilizadas por diversos pesquisadores para acompanhar a evolução de entidades discretas, tais como partículas, cristais, gotas, bolhas, células e espécies, ao longo do tempo em um determinado sistema associada a diversos fenômenos, como por exemplo nucleação, crescimento, aglomeração, quebra e coalescência (LIN *et al.*, 2016).

Essa abordagem considera que as partículas estão dispersas em uma fase contínua e elas podem interagir entre si e com a fase fluida. O vetor do estado das partículas é caracterizado pelas coordenadas externas (\mathbf{x}), que representam o espaço físico que elas ocupam, e as coordenadas internas, ($\boldsymbol{\xi}$) que são características das entidades como tamanho, idade, atividade química, ou qualquer outra característica importante para descrever o sistema. Já o vetor da fase contínua ($\mathbf{Y}(\mathbf{x}, t)$) contém as características do fluido que afetam o comportamento das partículas e é função apenas das coordenadas externas e do tempo (t) (YEOH *et al.*, 2014).

Ramkrishna (2000) escreveu a seguinte equação geral para o balanço populacional:

$$\frac{\partial f(\mathbf{x}, \boldsymbol{\xi}, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot \dot{\mathbf{X}}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t) f(\mathbf{x}, \boldsymbol{\xi}, t) + \nabla_{\boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t) f(\mathbf{x}, \boldsymbol{\xi}, t) = h(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t), \quad (3.22)$$

na qual $f(\mathbf{x}, \boldsymbol{\xi}, t)$ é a função densidade de probabilidade de número, $\nabla_{\mathbf{x}} \cdot$ representa o divergente sobre as coordenadas externas, $\dot{\mathbf{X}}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t) f(\mathbf{x}, \boldsymbol{\xi}, t)$, a taxa de mudança das coordenadas externas, $\nabla_{\boldsymbol{\xi}} \cdot$, o divergente sobre as coordenadas internas, $\dot{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t) f(\mathbf{x}, \boldsymbol{\xi}, t)$, a taxa de mudança das coordenadas internas e $h(\mathbf{x}, \boldsymbol{\xi}, \mathbf{Y}, t)$, o nascimento e morte das partículas.

O primeiro termo do lado esquerdo da Equação (3.22) refere-se ao termo transiente, o segundo termo, a advecção nas coordenadas externas e o terceiro, a advecção

nas coordenadas internas que está associada aos fenômenos contínuos como nucleação e crescimento superficial. Já o termo do lado direito está associado com fenômenos discretos de nascimento e morte das partículas por agregação e quebra, por exemplo.

Como para este estudo a variável interna de interesse é apenas o diâmetro da partícula, ξ é um escalar e a função densidade de probabilidade de número $f(\mathbf{x}, \xi, t)$ representa o número de partículas que um determinado diâmetro possui, ou seja, a distribuição do tamanho de partícula.

Considerando que não acontecesse os fenômenos de nucleação e crescimento das partículas durante o escoamento no ciclone, o terceiro termo da Equação (3.22), relacionado com a advecção nas coordenadas internas, pode ser desconsiderado. Considerando ainda que o nascimento e morte das partículas por quebra e coagulação podem ser desconsiderados, tem-se a Equação (3.22) simplificada como:

$$\frac{\partial f(\mathbf{x}, \xi, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot \dot{\mathbf{X}}(\mathbf{x}, \xi, \mathbf{Y}, t) f(\mathbf{x}, \xi, t) = 0. \quad (3.23)$$

Como a PBE é uma equação integro-diferencial com dependência temporal e espacial, muitas vezes sua solução analítica é inviável, tornando-se necessário a utilização de métodos numéricos. Dentre estes métodos, pode-se citar o método de Monte Carlo, das classes, e de momentos. O método de Monte Carlo utiliza uma abordagem estocástica e é bem flexível, entretanto ele demanda muito esforço computacional sendo inadequado para simulações de fluidodinâmica computacional. O método das classes é baseado na discretização da função densidade numérica em um número finito de classes, de modo que para cada uma das classes é adotado um valor constante para as variáveis internas. A grande desvantagem desse método é que se houver uma variação muito grande da função densidade numérica durante o processo seria necessário utilizar muitas classes para abranger toda a distribuição, o que demandaria muito esforço computacional e, além disso, muitas vezes não há como prever essa variação.

Por outro lado, pelo método dos momentos, é possível representar características da função densidade de probabilidade a partir de um conjunto de momentos de baixa ordem. Os momentos de uma distribuição são definidos pela Equação 2.9 e a partir deles pode-se calcular as medidas de dispersão, assimetria e forma de uma PDF. Contudo, o cálculo dos momentos de baixa ordem envolve momentos de ordem superior, surgindo assim um problema de fechamento. O método de quadratura baseado em momentos (*Quadrature Method of Moments* - QMOM) permite resolver o problema de fechamento para uma ampla variedade de problemas. Neste método, um momento de ordem k pode ser aproximado por uma quadratura Gaussiana, tal que:

$$M_k = \sum_{q=1}^N w_q \xi_q^k. \quad (3.24)$$

na qual w_q e ξ_q são os pesos e as abscissas da quadratura e N é o número de termos utilizados (pontos de quadratura).

No entanto, como o QMOM acompanha os momentos, há a necessidade de a cada iteração calcular os pesos e abscissas das quadraturas a partir dos momentos e ao final recalculá-los, o que resulta num alto custo computacional. Em contrapartida, o método de quadratura direta baseado em momentos (*direct quadrature method of moments* – DQMOM) acompanha diretamente os pesos e as abscissas, estando associado, então, um menor custo computacional e, assim, será utilizado neste trabalho.

3.1.5.1 Método de Quadratura Direta Baseado em Momentos

A ideia principal do DQMOM é escrever as equações de transporte diretamente com os pesos e as abscissas geradas pela aproximação das quadraturas. Deste modo, a função densidade numérica utilizando o DQMOM pode ser escrita como (NIEMI, 2012):

$$f(\mathbf{x}, \xi, t) = \sum_{q=1}^N w_q(\mathbf{x}, t) \delta[\xi - \xi_q], \quad (3.25)$$

na qual δ é a função delta de Kronecker e w_q e ξ_q são os pesos e abscissas dos N pontos de quadratura, respectivamente.

Substituindo a Equação (3.25) na Equação (3.23) e depois de algumas manipulações é possível escrever as equações de transporte para os pesos e abscissas das quadraturas como:

$$\frac{\partial w_q}{\partial t} + \nabla \cdot (\mathbf{v}_q w_q) = 0, \quad (3.26)$$

$$\frac{\partial \xi_q w_q}{\partial t} + \nabla \cdot (\mathbf{v}_q \xi_q w_q) = 0, \quad (3.27)$$

nas quais \mathbf{v}_q é o vetor velocidade de cada ponto de quadratura.

O primeiro trabalho para implementação do DQMOM em um escoamento gás-sólido foi o de Fan *et al.* (2004). Neste estudo, os autores expressam a fração volumétrica de cada ponto de quadratura utilizando os pesos e as abscissas. De modo que, para partículas esféricas, tem-se:

$$\alpha_q = \frac{\pi}{6} w_q \xi_q^3. \quad (3.28)$$

Assim, pode-se reescrever as Equações (3.26) e (3.27) como função de α_q e ξ_q :

$$\frac{(\partial \alpha_q \rho_q)}{\partial t} + \nabla \cdot (\mathbf{v}_q \alpha_q \rho_q) = 0, \quad (3.29)$$

$$\frac{(\partial \alpha_q \xi_q \rho_q)}{\partial t} + \nabla \cdot (\mathbf{v}_q \alpha_q \xi_q \rho_q) = 0. \quad (3.30)$$

Pode-se observar que a Equação (3.30) é uma combinação linear da Equação (3.29) e não necessita ser resolvida. Isto se deve ao fato de que neste trabalho foi considerado que não há nucleação, crescimento ou nascimento ou morte de partículas por quebra e coagulação e, portanto, a abscissa ξ_q permanece contante. Deste modo, a equação final do balanço populacional usando DQMOM se resume a Equação (3.29). Embora a velocidade das partículas (quadraturas) pudesse ser tratada como uma variável interna, neste trabalho aptou-se por utilizar a abordagem de mistura apresentada na próxima seção.

3.1.5.2 Abordagem de Mistura

A formulação de mistura descrita nessa seção foi desenvolvida por Niemi (2012) baseada na proposta por Manninen *et al.* (1996). No entanto, como neste trabalho a fase sólida foi considerada invíscida e foi desprezada a sua turbulência, algumas simplificações propostas por Niemi (2012) não foram necessárias. O ponto de partida é escrever as equações adicionais de momento para cada um dos N pesos. Assim, o conjunto de equações para o ponto de quadratura q resulta em,

$$\frac{\partial(\alpha_q \rho_q)}{\partial t} + \nabla \cdot (\alpha_q \rho_q \mathbf{v}_q) = 0, \quad (3.31)$$

$$\begin{aligned} \frac{\partial(\alpha_q \rho_q \mathbf{v}_q)}{\partial t} + \nabla \cdot (\alpha_q \rho_q \mathbf{v}_q \mathbf{v}_q) = & -\alpha_q \nabla p + \alpha_q \rho_q \mathbf{g} + K_{gs}(\mathbf{v}_g - \mathbf{v}_q) \\ & + \sum_{p \neq q}^N K_{pq}(\mathbf{v}_p - \mathbf{v}_q), \end{aligned} \quad (3.32)$$

na qual K_{pq} representa o coeficiente de arraste entre as fases sólidas p e q . Este arraste está associado à colisão entre os pontos de quadraturas, os quais representam partículas de diâmetros diferentes que possuem velocidades distintas. Assim, este coeficiente foi calculado utilizando o modelo de Syamlal (1987):

$$K_{pq} = \frac{3(1 + e_{pq})(\frac{\pi}{2} + C_{fr} \frac{\pi^2}{8}) \alpha_p \rho_p \alpha_q \rho_q (\xi_p + \xi_q)^2 g_{0,pq}}{2\pi(\rho_p \xi_p^3 + \rho_q \xi_q^3)} |\mathbf{v}_p - \mathbf{v}_q|, \quad (3.33)$$

na qual e_{pq} é o coeficiente de restituição, C_{fr} é o coeficiente de fricção e $g_{0,pq}$ é a função de distribuição radial, dada por:

$$g_{0,pq} = \frac{1}{\alpha_g} + \frac{3\xi_p \xi_q}{\alpha_g^2 (\xi_p + \xi_q)} \sum_{i=1}^N \left(\frac{\alpha_i}{\xi_i} \right). \quad (3.34)$$

Na abordagem de mistura, a ideia principal é somar as contribuições de todas as fases obtendo uma equação única para a mistura. Deste modo a equação do balanço populacional para mistura pode ser escrita como:

$$\frac{\partial \sum_{q=1}^N (\alpha_q \rho_q)}{\partial t} + \nabla \cdot \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_q) = 0. \quad (3.35)$$

Assumindo que a densidade seja uniforme para todas as partículas, a equação do balanço populacional pode ser simplificada a,

$$\frac{\partial(\alpha_s \rho_s)}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s) = 0, \quad (3.36)$$

de modo que a fração volumétrica e a velocidade podem ser definidas como:

$$\alpha_s = \sum_{q=1}^N \alpha_q, \quad (3.37)$$

$$\mathbf{v}_s = \frac{\sum_{q=1}^N \alpha_q \mathbf{v}_q}{\alpha_s}. \quad (3.38)$$

A equação do momento para a mistura também é a somatória de todas as fases:

$$\begin{aligned} \frac{\partial \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_q)}{\partial t} + \nabla \cdot \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_q \mathbf{v}_q) = & - \sum_{q=1}^N \alpha_q \nabla p + \sum_{q=1}^N \alpha_q \rho_q \mathbf{g} + \sum_{q=1}^N K_{gq} (\mathbf{v}_g - \mathbf{v}_q) \\ & + \sum_{q=1}^N \sum_{p \neq q}^N K_{pq} (\mathbf{v}_p - \mathbf{v}_q). \end{aligned} \quad (3.39)$$

Note que o ultimo termo da Equação (3.39) é igual a zero uma vez que ele terá sinais contrários para os pontos de quadratura p e q . O termo mais difícil para simplificar é o segundo termo, mas usando a definição de velocidade de deslizamento

$$\mathbf{v}_{qs} = \mathbf{v}_q - \mathbf{v}_s, \quad (3.40)$$

este termo pode ser escrito como:

$$\nabla \cdot \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_q \mathbf{v}_q) = \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s \mathbf{v}_s) + \nabla \cdot \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_{qs} \mathbf{v}_{qs}). \quad (3.41)$$

na qual a tensão difusiva devido a velocidade \mathbf{v}_{qs} pode ser definidas como:

$$\boldsymbol{\tau}_{Ds} = - \sum_{q=1}^N (\alpha_q \rho_q \mathbf{v}_{qs} \mathbf{v}_{qs}), \quad (3.42)$$

Assim, a Equação (3.39) pode ser simplificada como:

$$\frac{\partial(\alpha_s \rho_s \mathbf{v}_s)}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s \mathbf{v}_s) = -\alpha_s \nabla p + \nabla \cdot (\boldsymbol{\tau}_{Ds}) + \alpha_s \rho_s \mathbf{g} + K_{gs} (\mathbf{v}_g - \mathbf{v}_s), \quad (3.43)$$

de modo que:

$$K_{gs} (\mathbf{v}_g - \mathbf{v}_s) = \sum_{q=1}^N K_{gq} (\mathbf{v}_g - \mathbf{v}_q), \quad (3.44)$$

Reescrevendo as Equações (3.32) e (3.43) na forma não conservativa tem-se:

$$\begin{aligned} \alpha_q \rho_q \frac{\partial(\mathbf{v}_q)}{\partial t} + \alpha_q \rho_q (\mathbf{v}_q \cdot \nabla) \mathbf{v}_q = & -\alpha_q \nabla p + \alpha_q \rho_q \mathbf{g} + K_{gq}(\mathbf{v}_g - \mathbf{v}_q) \\ & + \sum_{p \neq q}^N K_{pq}(\mathbf{v}_p - \mathbf{v}_q), \end{aligned} \quad (3.45)$$

$$\alpha_s \rho_s \frac{\partial(\mathbf{v}_s)}{\partial t} + \alpha_s \rho_s (\mathbf{v}_s \cdot \nabla) \mathbf{v}_s = -\alpha_s \nabla p + \nabla \cdot (\boldsymbol{\tau}_{Ds}) + \alpha_s \rho_s \mathbf{g} + K_{gs}(\mathbf{v}_g - \mathbf{v}_s). \quad (3.46)$$

Igualando o gradiente de pressão do gás em ambas as equações e considerando que $\rho_q = \rho_s$, depois de algumas manipulações pode se obter:

$$\begin{aligned} K_{gq}(\mathbf{v}_g - \mathbf{v}_q) + \sum_{p \neq q}^N K_{pq}(\mathbf{v}_p - \mathbf{v}_q) = & \alpha_q \rho_s \frac{\partial \mathbf{v}_{qs}}{\partial t} + \alpha_q \rho_s ((\mathbf{v}_q \cdot \nabla) \mathbf{v}_q - (\mathbf{v}_s \cdot \nabla) \mathbf{v}_s) \\ & + \frac{\alpha_q}{\alpha_s} K_{gs}(\mathbf{v}_g - \mathbf{v}_s) + \frac{\alpha_q}{\alpha_s} \nabla \cdot (\boldsymbol{\tau}_{Ds}). \end{aligned} \quad (3.47)$$

Considerando que a diferença entre a velocidade dos pontos de quadratura e a da fase sólida não varia muito, pode-se assumir as seguintes aproximações:

$$\frac{\partial \mathbf{v}_{qs}}{\partial t} \approx 0, \quad (3.48)$$

$$(\mathbf{v}_q \cdot \nabla) \mathbf{v}_q \approx (\mathbf{v}_s \cdot \nabla) \mathbf{v}_s, \quad (3.49)$$

e que a tensão difusiva devido a velocidade \mathbf{v}_{qs} é muito menor que os de coeficientes de arraste, a Equação (3.47) torna-se:

$$K_{gq}(\mathbf{v}_g - \mathbf{v}_q) + \sum_{p \neq q}^N K_{pq}(\mathbf{v}_p - \mathbf{v}_q) = \frac{\alpha_q}{\alpha_s} K_{gs}(\mathbf{v}_g - \mathbf{v}_s). \quad (3.50)$$

Sendo assim, para encontrar a velocidade de cada quadratura, deve-se resolver o sistema de equações gerada pela Equação (3.50) utilizando um método numérico. Como as velocidades de cada ponto é aproximada devido às simplificações adotadas no seu processo de derivação, faz-se necessário corrigir a fração volumétrica para manter o balanço de massa em cada uma das células computacionais de modo a deixar a simulação estável e coerente após muitos passos de tempo.

3.1.5.3 Correção da Fração Volumétrica

Na metodologia da mistura, a soma das frações volumétricas obtida a partir dos pesos de cada ponto de quadratura deve ser igual a fração volumétrica da mistura como descrito na Equação (3.37). Entretanto, como a velocidade calculada para cada ponto de quadratura é apenas aproximada, é esperado que essa relação não se mantenha e de alguma forma a fração volumétrica tenha que ser corrigida. Niemi (2012) propôs que,

localmente, o somatório das novas frações volumétricas $\tilde{\alpha}_q$ são diferentes de α_s e assim devem ser normalizadas por:

$$\alpha_q = \tilde{\alpha}_q \frac{\alpha_s}{\sum_{q=1}^N \tilde{\alpha}_q}. \quad (3.51)$$

Assim, localmente, a Equação (3.37) passa a ser respeitada. No entanto, globalmente o volume total de uma fase q pode ter alterado e para corrigir o volume usa-se um fator de correção:

$$C_q = \frac{V_q}{\tilde{V}_q}. \quad (3.52)$$

na qual V_q é o real volume da fase q e \tilde{V}_q é o atual volume da fase q após a normalização. Assim, multiplica-se o fator de correção de cada fase pela fração volumétrica em cada ponto local. Naturalmente a equação (3.37) deixará de ser satisfeita e a normalização deve ser feita novamente. Esse processo iterativo é realizado até se obter um valor para a fração volumétrica dentro do erro esperado.

Niemi (2012), para ilustrar a correção da fração volumétrica, propôs a Figura 11 na qual há dois pontos de quadratura e três volumes de controle.

Inicial		Simulação		Normalização		Correção		Normalização
$\alpha_1=0,020$ $\alpha_2=0,010$ $\alpha_s=0,030$		$\alpha_1=0,025$ $\alpha_2=0,005$ $\alpha_s=0,028$		$\alpha_1=0,02333$ $\alpha_2=0,00467$ $\alpha_s=0,02800$		$\alpha_1=0,02317$ $\alpha_2=0,00472$ $\alpha_s=0,02800$		$\alpha_1=0,02326$ $\alpha_2=0,00474$ $\alpha_s=0,02800$
$\alpha_1=0,025$ $\alpha_2=0,015$ $\alpha_s=0,040$	→	$\alpha_1=0,020$ $\alpha_2=0,010$ $\alpha_s=0,035$	→	$\alpha_1=0,02333$ $\alpha_2=0,01167$ $\alpha_s=0,03500$	→	$\alpha_1=0,02317$ $\alpha_2=0,01179$ $\alpha_s=0,03500$	→	$\alpha_1=0,02318$ $\alpha_2=0,01182$ $\alpha_s=0,03500$
$\alpha_1=0,020$ $\alpha_2=0,020$ $\alpha_s=0,040$		$\alpha_1=0,020$ $\alpha_2=0,030$ $\alpha_s=0,047$		$\alpha_1=0,01880$ $\alpha_2=0,02820$ $\alpha_s=0,04700$		$\alpha_1=0,01867$ $\alpha_2=0,02850$ $\alpha_s=0,04700$		$\alpha_1=0,01858$ $\alpha_2=0,02842$ $\alpha_s=0,04700$
Total $\alpha_1=0,065$ $\alpha_2=0,045$ $\alpha_s=0,110$		Total $\alpha_1=0,065$ $\alpha_2=0,045$ $\alpha_s=0,110$		Total $\alpha_1=0,06547$ $\alpha_2=0,04453$ $\alpha_s=0,11000$		Total $\alpha_1=0,06500$ $\alpha_2=0,04500$ $\alpha_s=0,11000$		Total $\alpha_1=0,06506$ $\alpha_2=0,04494$ $\alpha_s=0,11000$

Figura 11 – Exemplo da correção da fração volumétrica (Adaptado de Niemi (2012))

3.1.6 Síntese da Modelagem Matemática

As equações básicas da modelagem matemática do ciclone com alta carga estão resumidas na Tabela 5. De modo que, os coeficientes de arraste gás-sólido e sólido-sólido

foram calculados seguindo os modelos de Gidaspow (1994) e Syamlal (1987), respectivamente, o diâmetro característico da fase sólida em cada célula computacional será calculado pela divisão do momento de terceira ordem pelo de segunda ordem calculados a partir dos pesos e abscissas e o modelo de turbulência utilizado foi o RSM proposto por Speziale *et al.* (1991).

Tabela 5 – Síntese das equações básicas da modelagem matemática que serão resolvidas na simulação numérica de ciclones com alta carga de sólidos

Nome	Equação
Equação da continuidade para i-ésima fase	$\frac{\partial(\alpha_i \rho_i)}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{v}_i) = 0$
Equação de quantidade de movimento para fase gasosa	$\frac{\partial(\alpha_g \rho_g \mathbf{v}_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{v}_g \mathbf{v}_g) = -\alpha_g \nabla p + \nabla \cdot (\boldsymbol{\tau}_g - \alpha_g \boldsymbol{\tau}_{Rg}) + \alpha_g \rho_g \mathbf{g} + K_{gs}(\mathbf{v}_s - \mathbf{v}_g)$
Equação de quantidade de movimento para fase particulada	$\frac{\partial(\alpha_s \rho_s \mathbf{v}_s)}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{v}_s \mathbf{v}_s) = -\alpha_s \nabla p + \alpha_s \rho_s \mathbf{g} + K_{gs}(\mathbf{v}_g - \mathbf{v}_s)$
Equações de balanço populacional para cada ponto de quadratura q	$\frac{\partial \alpha_q \rho_q}{\partial t} + \nabla \cdot (\mathbf{v}_q \alpha_q \rho_q) = 0$
Equação para o cálculo das velocidades de cada ponto de quadratura q	$K_{gq}(\mathbf{v}_g - \mathbf{v}_q) + \sum_{p \neq q}^N K'_{pq}(\mathbf{v}_p - \mathbf{v}_q) = \frac{\alpha_q}{\alpha_s} K_{gs}(\mathbf{v}_g - \mathbf{v}_s)$

3.2 SIMULAÇÃO NUMÉRICA

O primeiro passo da simulação numérica é a implementação e verificação do código para a modelagem proposta, seguido da sua aplicação em ciclones operando a alta carga de sólidos.

3.2.1 Implementação das Equações do Modelo

A implementação das equações do modelo ao código do *software* comercial Fluent é dado pelas funções definidas pelo usuário (*user define function* - UDF), sub-rotinas escritas em linguagem C que permitem customizar condições de contorno, ajustar valores calculados, modificar ou adicionar modelos, dentre outros usos (ANSYS, 2016).

No intuito de definir qual sub-rotina utilizar é necessário compreender a sequência em que os processos são executados pelo Fluent. A Figura 12 ilustra, através de um

fluxograma, a ordem de execução dos processos para o solver baseado na pressão em cada passo de tempo (ANSYS, 2016).

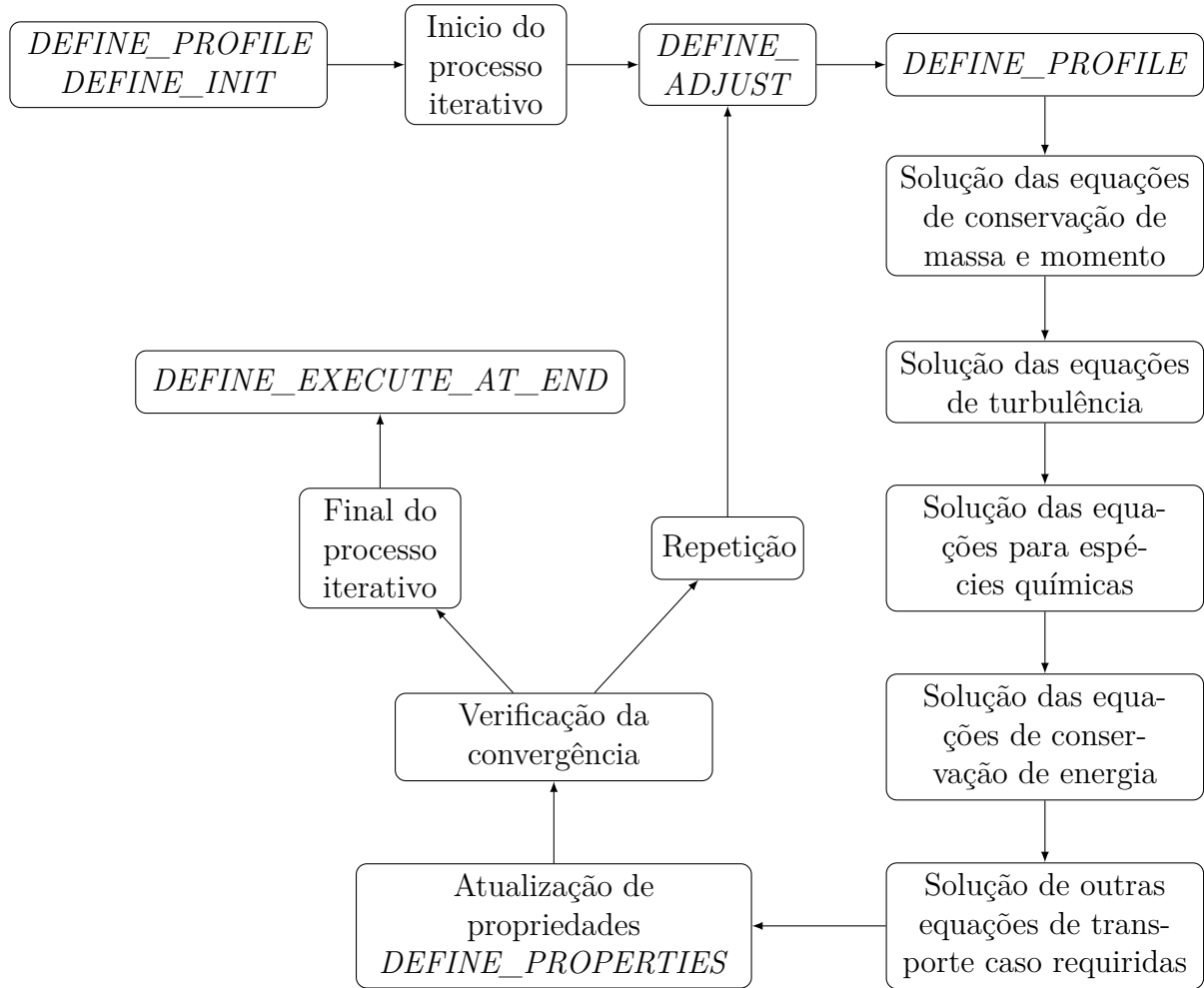


Figura 12 – Fluxograma de execução do processo do Fluent para o solver baseado na pressão em cada passo de tempo (Adaptado de ANSYS (2016))

Para implementar as equações resultantes da aplicação do DQMOM, representadas pela Equação (3.29), foi necessário obter a solução de N outras equações de transporte. Para isso, foi definido N escalares (*user-defined scalar* - UDS) no Fluent e para cada um destes escalares foram criadas sub-rotinas para cálculo dos termos convectivo e transiente da Equação (3.29), sendo elas, respectivamente, *DEFINE_UDS_FLUX* e *DEFINE_UDS_UNSTEADY*.

No entanto, a Equação (3.29) necessita das velocidades dos N pontos de quadratura. Assim, foram definidas $3N$ memórias (*user-defined memory* - UDM) para armazenar os valores das velocidades. Estas foram calculadas resolvendo o sistema não linear gerado pela equação (3.50) via método de Newton e eliminação de Gauss e acopladas ao Fluent por meio da sub-rotina *DEFINE_ADJUST*. Por fim, a fração volumétrica foi corrigida utilizando a sub-rotina *DEFINE_EXECUTE_AT_END* e o diâmetro característico foi especificado com a sub-rotina *DEFINE_PROPERTY*.

3.2.2 Verificação das Sub-rotinas

Com o intuito de verificar a implementação das sub-rotinas, foi reproduzido o estudo desenvolvido por Niemi (2012), no qual foram realizadas simulações em 2D de um CFB. A Figura 13 apresenta um esquema da geometria que ressalta as alturas de 23 cm, 40 cm, 80 cm e 120 cm, as quais foram utilizadas para comparar os resultados obtidos por este trabalho e os obtidos por Niemi (2012). Uma malha hexaédrica foi construída com hexaedros de 5 mm de lado, totalizando 49780 células. Ambas, geometria e malha, foram geradas utilizando o *software* comercial ICEM CFD 17.2.

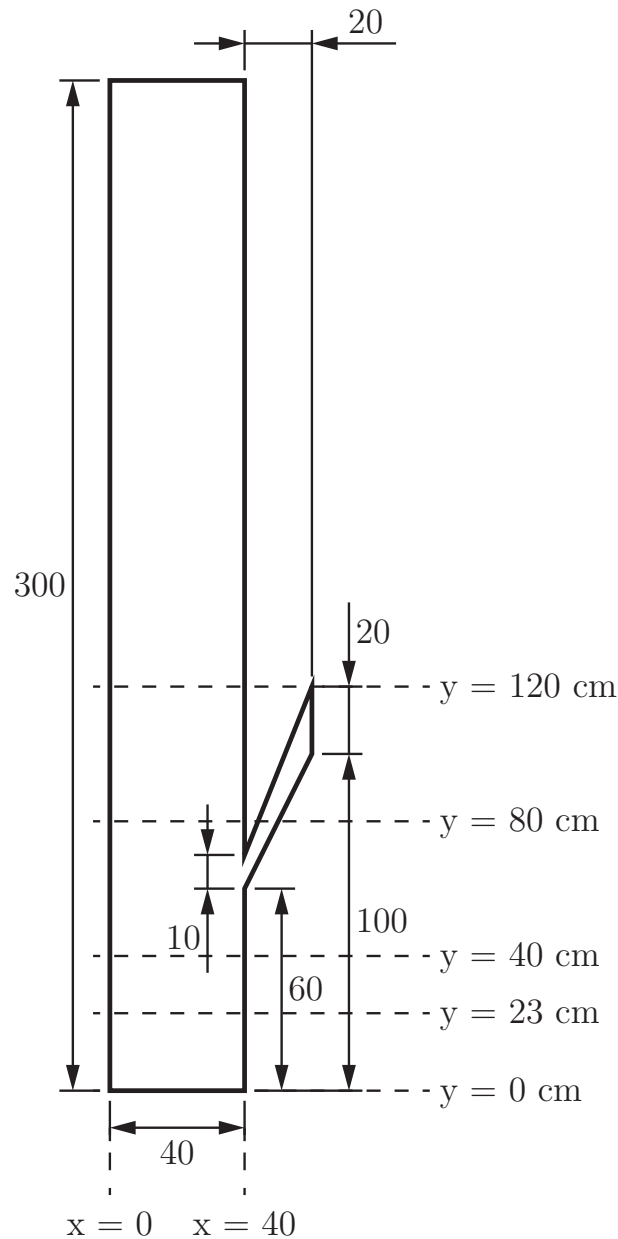


Figura 13 – Esquema da geometria do CFB em cm ressaltando as alturas de 23 cm, 40 cm, 80 cm e 120 cm (Adaptado de Niemi (2012))

A modelagem da fase sólida proposta por Niemi (2012) foi baseada na teoria

cinética de escoamento granular (*kinetic theory of granular flow* - KTGF) de modo que alguns modelos e parâmetros adicionais necessitam ser definidos. De forma a reproduzir o trabalho de Niemi (2012) os mesmos modelos e parâmetros foram utilizados e estão sintetizados na Tabela 6.

Tabela 6 – Modelos e parâmetros utilizados na modelagem de Niemi (2012)

Termo	Modelo
<i>Collisional viscosity</i>	Gidaspow (1994)
<i>Granular Viscosity</i>	Syamlal <i>et al.</i> (1993)
<i>Granular Bulk Viscosity</i>	Lun <i>et al.</i> (1984)
<i>Frictional viscosity</i>	Schaeffer (1987)
<i>Angle of internal friction</i>	30°
<i>Frictional Pressure</i>	Baseado na KTGF
<i>Frictional Modulus</i>	Derivado
<i>Friction Packing Limit, $\alpha_{s,min}$</i>	0,61
<i>Granular Conductivity</i>	Syamlal <i>et al.</i> (1993)
<i>Solid Pressure</i>	Lun <i>et al.</i> (1984)
<i>Radial Distribution</i>	Lun <i>et al.</i> (1984)
<i>Elasticity Modulus</i>	Derivado
<i>Packing Limit, $\alpha_{s,min}$</i>	0,63
<i>Drag Coefficient, K_{gs}</i>	Gidaspow (1994)
<i>Restitution coefficient, e_{ss}</i>	0,9
Turbulência	$k - \epsilon$ disperso

Foram utilizadas quatro condições de contorno distintas: entrada de gás no fundo do leito, saída na parte superior, entrada de sólido na lateral e paredes. A entrada de gás é retratada por um perfil de velocidade, apresentado na Figura 14, de modo a representar os oito bicos injetores presentes no aparato experimental. Esse perfil foi implementado no Fluent utilizando a sub-rotina *DEFINE_PROFILE* e o código está disponível no Apêndice B.1. Na saída foi usada a condição de pressão constante a 1 atm e para os pontos de quadratura foi considerado que o valor da fração volumétrica no centro da célula computacional adjacente à condição de contorno seria o valor da condição de contorno, essa condição de contorno foi implementada e se encontra disponível no Apêndice B.3. Como se trata de um leito fluidizado circulante, a entrada de sólidos tem que ser igual a quantidade de sólidos que sai do leito, assim, essa condição de contorno também foi implementada no Fluent e está à disposição no Apêndice B.2. Por fim, nas paredes foi utilizada a condição de não deslizamento (*no-slip boundary condition*) para o gás e para o sólido uma condição de deslizamento parcial com o coeficiente de especularidade (*specularity coefficient*) de 0,001.

Considerando a altura inicial do leito como 0.6 m, a fração volumétrica inicial utilizada na simulação foi 0,31922. Niemi (2012) utilizou dois pontos de quadratura com fração volumétrica de 0,78 e 0,22 e abcissa de 255 μm e 416 μm , respectivamente, para

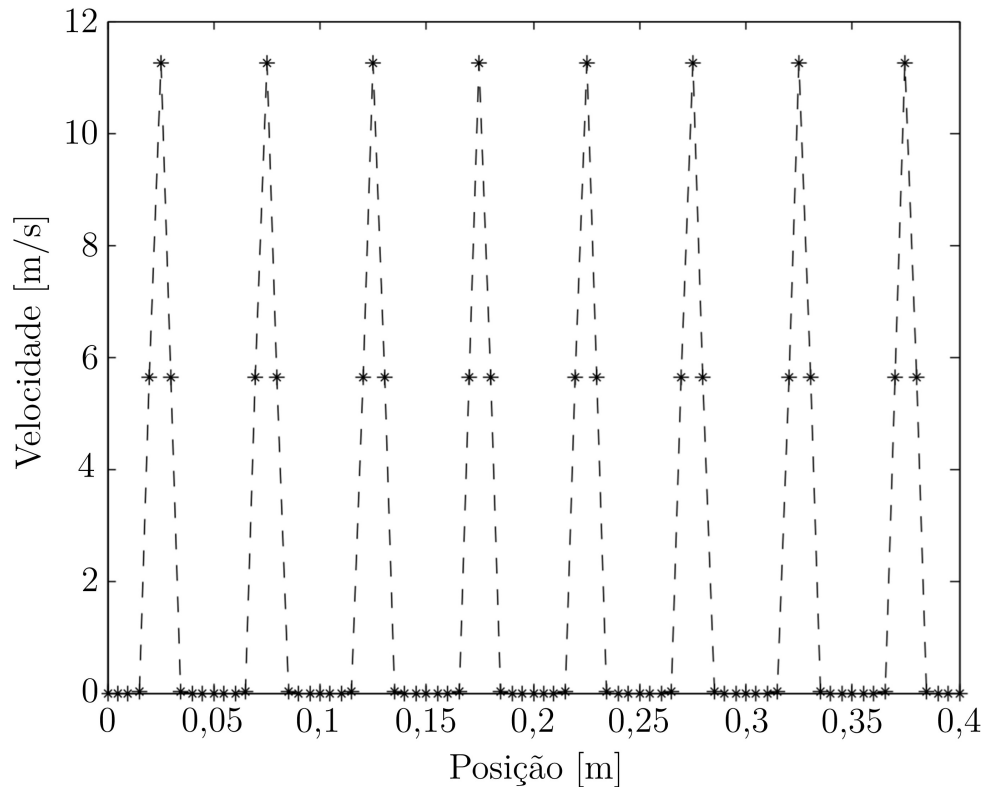


Figura 14 – Perfil de velocidade do gás utilizado como condição de contorno (Adaptado de Niemi (2012))

representar uma distribuição de tamanho de partículas com diâmetro médio de Sauter de $279 \mu m$. Os esquemas e algoritmos numéricos relevantes utilizados na simulação estão resumidos na Tabela 7. Foram simulados 60 s de tempo real de modo a atingir o estado quasi-estacionário a partir do qual foram calculados os perfis fluidodinâmicos médios durante 120 s. Todas as simulações foram realizadas em um *cluster* computacional utilizando um nó com 48 GB de memória RAM e doze unidades de processamento em *sockets* Intel Xeon X5675 de 3,07 GHz.

Tabela 7 – Esquemas e algoritmos numéricos utilizados na simulação realizada por Niemi (2012)

Termo	Modelo
Acoplamento Pressão-Velocidade	<i>Phase Coupled SIMPLE</i>
Gradiente	<i>Green-Gauss Cell Based</i>
Momento	QUICK
Fração Volumétrica	QUICK
Temperatura Granular	<i>First Order Upwind</i>
Energia Cinética Turbulenta	<i>First Order Upwind</i>
Taxa de Dissipação Turbulenta	<i>First Order Upwind</i>

3.2.3 Avaliação do Número de Quadraturas para CFB

De modo a complementar o estudo realizado por Niemi (2012) e entender como o número de quadraturas pode influenciar nos resultados foram propostos dois testes. O primeiro consiste em determinar o número mínimo de momentos necessários para reconstruir a função densidade de probabilidade utilizando os métodos de máxima entropia e *spline* cúbica e o outro visa analisar o efeito do aumento do número de pontos de quadraturas sobre o comportamento fluidodinâmico do CFB.

Para realizar estes dois testes, foi proposta uma função densidade de probabilidade de número log-normal com diâmetro médio de $264,97 \mu m$, desvio padrão de $109,64 \mu m$ e diâmetro médio de Sauter de $355 \mu m$. A Figura 15 apresenta a forma da PSD inicial utilizada. Com base nessa distribuição foram calculados os seus doze primeiros momentos que estão representados na Tabela 8. Assim, utilizando estes momentos e algoritmos descritos nas seções 3.2.5.1 e 3.2.5.2 para os métodos de máxima entropia e *spline* cúbica, respectivamente, a distribuição proposta foi reconstruída utilizando três, quatro, cinco, seis, sete, oito, nove e dez momentos.

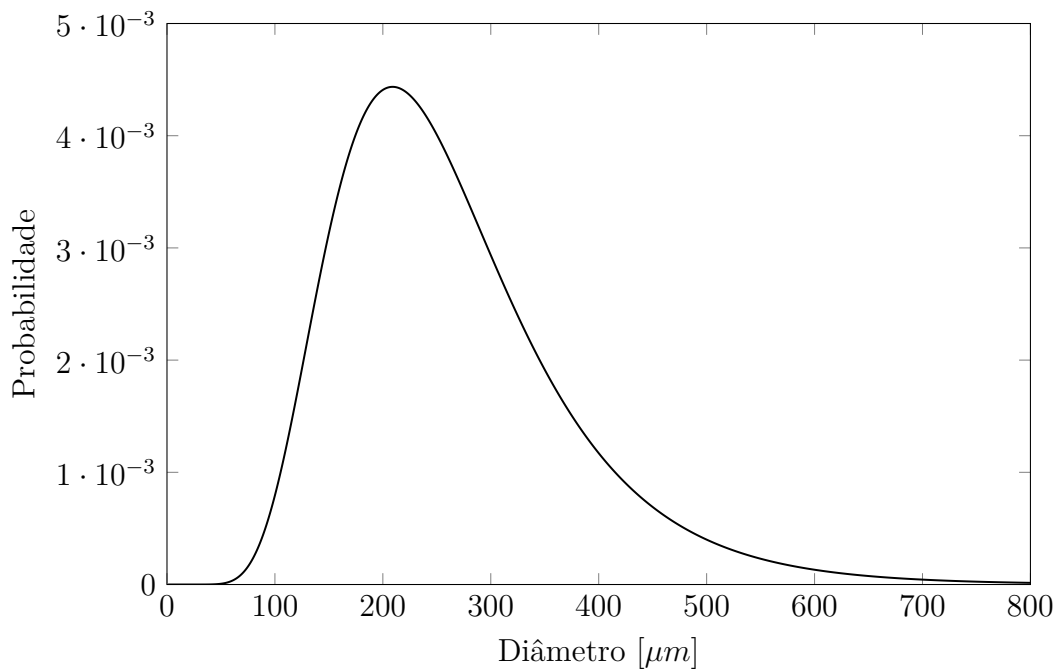


Figura 15 – Distribuição do tamanho de partícula utilizada para avaliar a influência do número de pontos de quadraturas

Ademais, com o intuito de avaliar o efeito do número de pontos de quadratura no comportamento fluidodinâmico do leito fluidizado circulante, foram calculados os pesos e abscissas para os casos com dois, três, quatro, cinco e seis pontos de quadratura a partir dos momentos apresentados na Tabela 8 utilizando algoritmo Produto-Diferença descrito na seção 3.2.4. A Tabela 9 apresenta os pesos e abscissas para os casos com dois, três,

Tabela 8 – Momentos da distribuição utilizada para avaliar a influência do número de pontos de quadraturas

Ordem do momento	Valor
0	1
1	$2,6405 \cdot 10^2 \mu m$
2	$8,1162 \cdot 10^4 \mu m^2$
3	$2,8827 \cdot 10^7 \mu m^3$
4	$1,1694 \cdot 10^{10} \mu m^4$
5	$5,3336 \cdot 10^{12} \mu m^5$
6	$2,6862 \cdot 10^{15} \mu m^6$
7	$1,4671 \cdot 10^{18} \mu m^7$
8	$8,5494 \cdot 10^{20} \mu m^8$
9	$5,2433 \cdot 10^{23} \mu m^9$
10	$3,3476 \cdot 10^{26} \mu m^{10}$
11	$2,2061 \cdot 10^{29} \mu m^{11}$

quadro, cinco e seis pontos de quadratura, além de apresentar a fração volumétrica de cada ponto de quadratura calculada pela Equação (3.28).

Tabela 9 – Os pesos (w), as abscissas (ξ) e as frações volumétricas (α) iniciais do CFB de cada ponto de quadratura (q) para os casos com o total de N pontos de quadraturas

		q					
		1	2	3	4	5	6
$N = 2$	w	0,742	0,258	-	-	-	-
	α	0,209	0,791	-	-	-	-
	$\xi [\mu m]$	201	446	-	-	-	-
$N = 3$	w	0,465	0,491	0,044	-	-	-
	α	0,073	0,596	0,331	-	-	-
	$\xi [\mu m]$	166	327	600	-	-	-
$N = 4$	w	0,269	0,559	0,160	0,012	-	-
	α	0,026	0,349	0,492	0,133	-	-
	$\xi [\mu m]$	141	262	445	689	-	-
$N = 5$	w	0,148	0,504	0,291	0,052	0,005	-
	α	0,010	0,185	0,457	0,283	0,065	-
	$\xi [\mu m]$	123	219	356	541	733	-
$N = 6$	w	0,107	0,447	0,344	0,089	0,012	0,001
	α	0,006	0,130	0,401	0,341	0,120	0,002
	$\xi [\mu m]$	115	202	321	477	650	773

3.2.4 Algoritmo Produto-Diferença

O algoritmo produto-diferença é um método para calcular os pesos e abcissas de N pontos de quadratura a partir de $2N$ momentos. Esse procedimento foi primeiramente apresentado por Gordon (1968) e o primeiro passo é definir uma matriz $P(i, j)$ de ordem $(2N + 1) \times (2N + 1)$ na qual a primeira coluna pode ser representada por,

$$P(i, 1) = \begin{cases} 1, & \text{se } i = 1, \\ 0, & \text{se } i = 2, \dots, 2N + 1, \end{cases} \quad (3.53)$$

a segunda coluna contém os momentos com sinais alternados:

$$P(i, 2) = (-1)^{i-1} M_{i-1} \quad i = 1, 2, \dots, 2N, \quad (3.54)$$

e as demais colunas podem ser escritas como:

$$P(i, j) = P(1, j-1)P(i+1, j-2) - P(1, j-2)P(i+1, j-1), \quad (3.55)$$

na qual j varia de 3 até $2N + 1$ colunas e i varia entre 1 e $2N + 2 - j$. Para simplificar pode-se normalizar os momentos pelo momento de ordem zero. Assim tomando como exemplo $N = 2$ a matriz P pode ser escrita como:

$$P(i, j) = \begin{bmatrix} 1 & 1 & M_1 & M_2 - M_1^2 & M_1 M_3 - M_2^2 \\ 0 & -M_1 & -M_2 & -M_3 + M_1 M_2 & 0 \\ 0 & M_2 & M_3 & 0 & 0 \\ 0 & -M_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.56)$$

Após obter a matriz P , deve-se calcular o vetor β com N dimensões por,

$$\beta(j) = \begin{cases} 0, & \text{se } j = 1, \\ \frac{P(1, j+1)}{P(1, j)P(1, j-1)}, & \text{se } j = 2, \dots, 2N. \end{cases} \quad (3.57)$$

A partir do vetor β , deve-se calcular a diagonal principal a_j e a co-diagonal b_j da matriz simétrica tridiagonal T por,

$$a_j = \beta(2j) + \beta(2j-1) \quad j = 1, \dots, N, \quad (3.58)$$

$$b_j = -\sqrt{\beta(2j)\beta(2j+1)} \quad j = 1, \dots, N-1. \quad (3.59)$$

Assim, a matriz T pode ser representada como:

$$T(i, j) = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & \ddots & & \\ & \ddots & \ddots & b_{N-1} & \\ & & b_{N-1} & a_N & \end{bmatrix}. \quad (3.60)$$

Por fim, para encontrar os valores dos pesos e das abcissas deve-se calcular os autovalores e autovetores da matriz T . As abcissas são os autovalores e os pesos são o primeiro componente dos autovetores ao quadrado. Caso a matriz P tenha sido normalizada pelo momento de ordem zero, os pesos são a multiplicação do primeiro componente dos autovetores ao quadrado pelo momento de ordem zero.

3.2.5 Reconstrução de Função Densidade de Probabilidade

Em busca de reconstruir a curva de eficiência de coleta do ciclone, se faz necessário saber a PDF que está deixando o ciclone pelo *vortex finder*. Sendo assim, precisa-se reconstruir a função densidade de probabilidade a partir dos momentos calculados pelos pontos de quadratura após a simulação numérica. Teoricamente, uma reconstrução perfeita requer um número infinito de momentos o que inviabilizaria completamente a utilização de soluções a partir de métodos dos momentos (LEBAZ *et al.*, 2016).

A saída mais simples é aproximar a PDF a uma distribuição conhecida, presumindo sua forma e, assim, com apenas três momentos pode-se reconstruí-la. Contudo, é difícil prever a forma da distribuição após um processo. Deste modo, na literatura, existem alguns métodos para reconstruir a função densidade de probabilidade, dentre eles, pode-se citar o método de máxima entropia e *spline* cúbica que serão abordados nas seções 3.2.5.1 e 3.2.5.2, respectivamente.

3.2.5.1 Máxima entropia

Mohammad-Djafari (1991) desenvolveu um método para reconstruir a função densidade de probabilidade por minimização da entropia de Shannon (1948) sujeita as restrições dos momentos. O problema de otimização, então, se resume a:

$$\begin{aligned} &\text{maximizar} && H = - \int f(x) \ln(f(x)) dx \\ &\text{sujeito a} && M_k = \int x^k f(x) dx \quad k = 0, \dots, 2N - 1. \end{aligned} \quad (3.61)$$

A solução clássica para esse problema de otimização é:

$$f(x) = e^{-\sum_{i=0}^{2N-1} \lambda_i x^i}, \quad (3.62)$$

na qual $\boldsymbol{\lambda} = [\lambda_0, \dots, \lambda_{2N-1}]^T$ são os multiplicadores de Lagrange, que são obtidos resolvendo o seguinte sistema de $2N$ equações não lineares:

$$\int x^k e^{-\sum_{i=0}^{2N-1} \lambda_i x^i} = M_k \quad k = 0, \dots, 2N - 1 \quad (3.63)$$

Esse sistema pode ser resolvido pelo método de Newton que necessita de um chute inicial para os multiplicadores de Lagrange. Uma estimativa, num intervalo $[x_1, \dots, x_f]$ é pressupor uma distribuição uniforme, cujo os multiplicadores de Lagrange são $\boldsymbol{\lambda}^0 = [\ln(x_f - x_1), 0, \dots, 0]^T$ para uma distribuição normalizada, ou seja, $M_0 = 1$.

Mohammad-Djafari (1991) disponibilizou o código para o *software* comercial MATLAB. Foram feitas pequenas modificações para salvar os dados da reconstrução e computar o tempo de execução do programa. Esse código está no Anexo A.1 e pode ser executado no *software* livre GNU Octave.

3.2.5.2 Spline cúbica

John *et al.* (2007) propuseram essa técnica de reconstrução da função densidade de probabilidade. Baseada em funções polinomiais de terceiro grau entre dois pontos de um subintervalo, considerando que a função densidade de probabilidade está contida no intervalo $[x_1, \dots, x_f]$. Este intervalo pode ser subdividido em n subintervalos igualmente espaçados, de modo que pode-se escrever a função polinomial de terceiro grau (s) como:

$$s_i(x) = \sum_{j=0}^3 s_{ij}(x - x_i)^j, \quad x \in [x_i, x_{i+1}], \quad i = 1, \dots, n. \quad (3.64)$$

Deste modo, há $4n$ coeficientes s_{ij} que deverão ser determinados resolvendo um sistema linear mal condicionado. Para isso, é necessário definir também a primeira e segunda derivadas da função polinomial:

$$s'_i(x) = \sum_{j=1}^3 j s_{ij}(x - x_i)^{j-1}, \quad x \in [x_i, x_{i+1}], \quad i = 1, \dots, n. \quad (3.65)$$

$$s''_i(x) = \sum_{j=2}^3 j(j-1) s_{ij}(x - x_i)^{j-2}, \quad x \in [x_i, x_{i+1}], \quad i = 1, \dots, n. \quad (3.66)$$

Como a função está contida no intervalo $[x_1, \dots, x_f]$, fora desse intervalo ela é considerada como sendo aproximadamente zero e, ainda, que a transição da função nos contornos é suave, ou seja, a primeira e a segunda derivadas da função polinomial são zero. Desta forma, para x_1 utilizando as Equações (3.64),(3.65),(3.66) tem-se:

$$s_1(x_1) = s_{10} = 0, \quad s'_1(x_1) = s_{11} = 0, \quad s''_1(x_1) = 2s_{12} = 0, \quad (3.67)$$

reescrevendo na forma matricial:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.68)$$

De forma análoga para x_f ou x_{n+1} tem-se:

$$\begin{bmatrix} 1 & (x_{n+1} - x_n) & (x_{n+1} - x_n)^2 & (x_{n+1} - x_n)^3 \\ 0 & 1 & 2(x_{n+1} - x_n) & 3(x_{n+1} - x_n)^2 \\ 0 & 0 & 2 & 6(x_{n+1} - x_n) \end{bmatrix} \begin{bmatrix} s_{n0} \\ s_{n1} \\ s_{n2} \\ s_{n3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.69)$$

Para os $n-1$ pontos internos, a condição de continuidade tem que ser satisfeita, ou seja, para a função polinomial tem-se: $s_i(x_{i+1}) - s_{i+1}(x_i) = 0$, de modo que:

$$\begin{bmatrix} 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 & (x_{i+1} - x_i)^3 & -1 \end{bmatrix} \begin{bmatrix} s_{i0} \\ s_{i1} \\ s_{i2} \\ s_{i3} \\ s_{(i+1)0} \end{bmatrix} = 0. \quad (3.70)$$

Analogamente, para a primeira derivada tem-se $s'_i(x_{i+1}) - s'_{i+1}(x_i) = 0$ de forma que:

$$\begin{bmatrix} 0 & 1 & 2(x_{i+1} - x_i) & 3(x_{i+1} - x_i)^2 & -1 \end{bmatrix} \begin{bmatrix} s_{i0} \\ s_{i1} \\ s_{i2} \\ s_{i3} \\ s_{(i+1)1} \end{bmatrix} = 0. \quad (3.71)$$

De forma semelhante, para segunda derivada, $s''_i(x_{i+1}) - s''_{i+1}(x_i) = 0$, tem-se:

$$\begin{bmatrix} 0 & 1 & 2 & 6(x_{i+1} - x_i) & -2 \end{bmatrix} \begin{bmatrix} s_{i0} \\ s_{i1} \\ s_{i2} \\ s_{i3} \\ s_{(i+1)2} \end{bmatrix} = 0. \quad (3.72)$$

Assim sendo, é possível escrever 3 equações para cada um dos $n+1$ pontos totalizando $3n+3$ equações. As equações restantes necessárias para a resolução do sistema advêm dos momentos. Assim, para um número arbitrário de momentos ($2N$), o número total de subintervalos (n) é:

$$n = 2N + 3. \quad (3.73)$$

Pode-se escrever as equações dos momentos como:

$$M_k = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} x^k s_i(x) dx = \sum_{i=1}^n \sum_{j=0}^3 s_{ij} \int_{x_i}^{x_{i+1}} x^k (x - x_i)^j dx, \quad k = 0, \dots, 2N-1. \quad (3.74)$$

Definindo os seguintes parâmetros:

$$I_1 = \frac{x_{i+1}^{k+1} - x_i^{k+1}}{k+1}, \quad I_2 = \frac{x_{i+1}^{k+2} - x_i^{k+2}}{k+2}, \quad I_3 = \frac{x_{i+1}^{k+3} - x_i^{k+3}}{k+3}, \quad I_4 = \frac{x_{i+1}^{k+4} - x_i^{k+4}}{k+4}, \quad (3.75)$$

e então reescrever a Equação (3.74) para $k = 0, \dots, 2N-1$ como:

$$M_k = \sum_{i=1}^n [I_1 s_{i0} + (I_2 - x_i I_1) s_{i1} + (I_3 - 2x_i I_2 + x_i^2 I_1) s_{i2} + (I_4 - 3x_i I_3 + 3x_i^2 I_2 - x_i^3 I_1) s_{i3}]. \quad (3.76)$$

Após a solução deste sistema, verifica-se se os valores da função no primeiro e último subintervalos são próximos de zero. Caso sejam o intervalo é redimensionado para o valor médio do subintervalo em questão. Isto acontece até ser encontrado o menor intervalo possível que contenha a função. Como se trata de reconstruir uma função densidade de probabilidade não podem haver valores negativos, assim, deve-se verificar a existência de valores negativos e caso tenham deve-se remover o menor valor singular da matriz dos coeficientes. Caso essas duas restrições sejam satisfeitas a reconstrução estará terminada.

Contudo, foi observado que essa técnica não permitia determinar picos estreitos. Assim, Souza *et al.* (2010) adaptaram esse algoritmo de modo a realocar os nós dos subintervalos para onde a segunda derivada troca de sinal e para os maiores valores da primeira derivada respeitando uma distância mínima entre nós. Dessa maneira, alterando o tamanho dos subintervalos foi possível aproximar a curva reconstruída da função densidade de probabilidade para picos estreitos. O código desenvolvido por John *et al.* (2007) e adaptado por Souza *et al.* (2010) para o *software* comercial MATLAB está disponível em <<http://www.lss.ovgu.de/lss/en/Infos/Downloads.html>>. Algumas adaptações foram realizadas no código para salvar os dados da reconstrução, determinar o tempo de execução do programa e também permitir executá-lo no *software* livre GNU Octave. O código adaptado se encontra no Anexo A.2.

3.2.6 Ciclone com Alta Carga

O estudo do ciclone de alta carga foi baseado no trabalho de Huang *et al.* (2018), o qual apresenta resultados numéricos e experimentais utilizando a geometria esquematizada na Figura 16.

Nos estudos experimentais e numéricos realizados pelos autores, foram utilizadas partículas com massa específica de $2900 \text{ kg} \cdot \text{m}^{-3}$ e que possuíam a distribuição mássica acumulada do tamanho de partículas representada na Figura 17.

As condições de contorno do caso escolhido para estudo foram de velocidade de $15 \text{ m} \cdot \text{s}^{-1}$ e fração volumétrica de $3,98 \cdot 10^{-5}$ na entrada e pressão ambiente na saída do *vortex finder*. Nas paredes foi utilizada a condição de não deslizamento para ambas as fases e foi considerado que o sólido que atinge o *hopper* é retirado do sistema. Esta última condição de contorno foi implementada utilizando a sub-rotina *DEFINE_EXECUTE_AT_END*.

Deste modo, primeiramente, foi realizado um estudo da incerteza numérica associada ao tamanho da malha e, posteriormente, foram avaliados os efeitos do número de pontos de quadratura no comportamento fluidodinâmico do ciclone, bem como, na determinação da curva de eficiência de coleta. Além disso, foi analisado o efeito do arraste nos parâmetros de desempenho do ciclone. Por fim, foram avaliados outros casos variando a

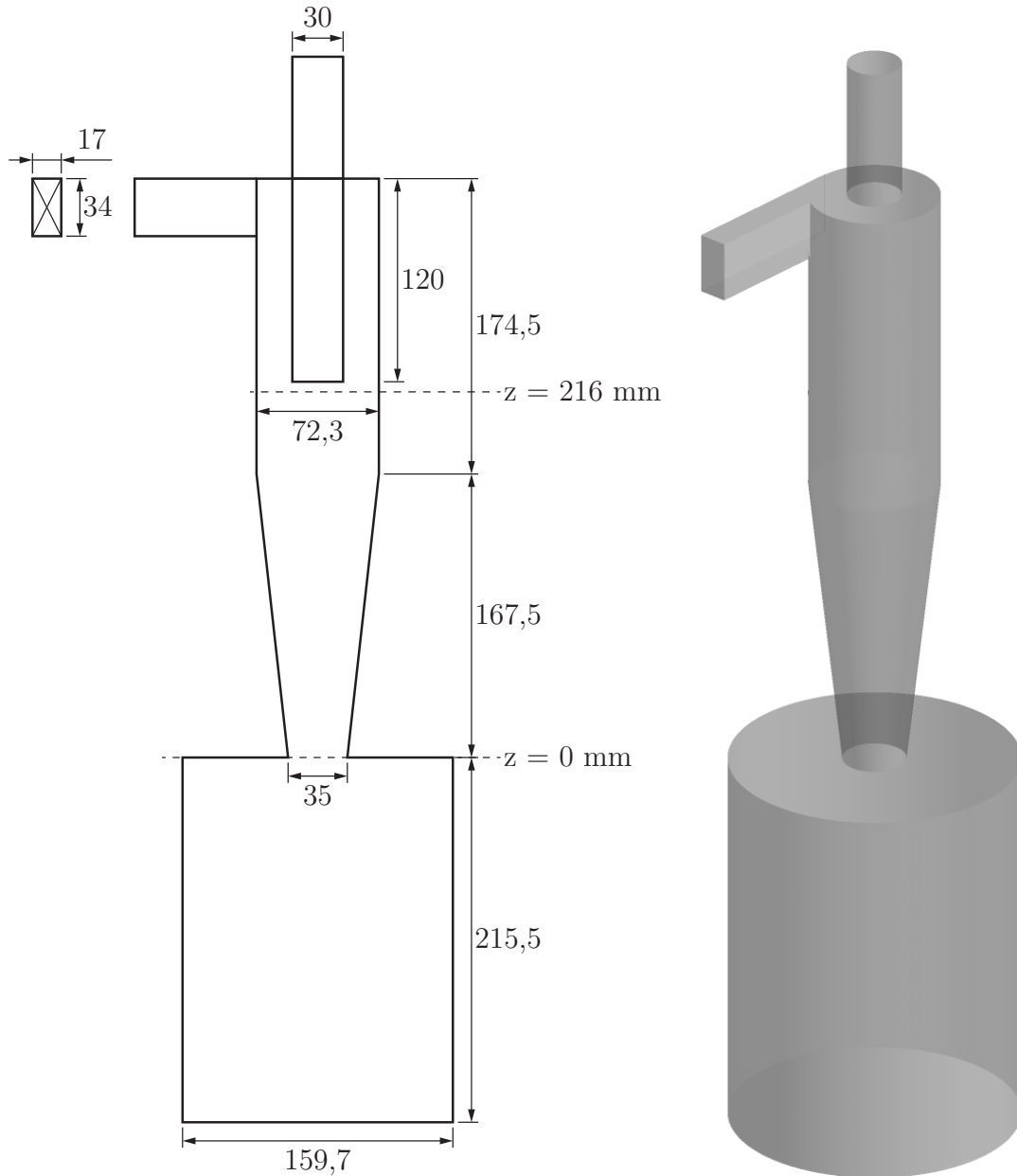


Figura 16 – Esquema das dimensões da geometria do ciclone em milímetros (Adaptado de Huang *et al.* (2018))

velocidade de entrada do ciclone e a carga de sólidos. Todas as simulações foram realizadas em um *cluster* computacional utilizando cinco nós com 132 GB de memória RAM e vinte e oito unidades de processamento em *sockets* Intel Xeon E5-2680 de 2,40 GHz.

3.2.6.1 Teste de independência de malhas

No intuito de avaliar a independência da malha sobre os resultados dos ciclones avaliados foi calculado o índice de convergência de grade (*grid convergence index* - GCI). O GCI foi proposto por Roache (1994) e é baseado na teoria geral da extrapolação de Richardson para estimar o erro do refinamento da malha e vem sendo utilizada em

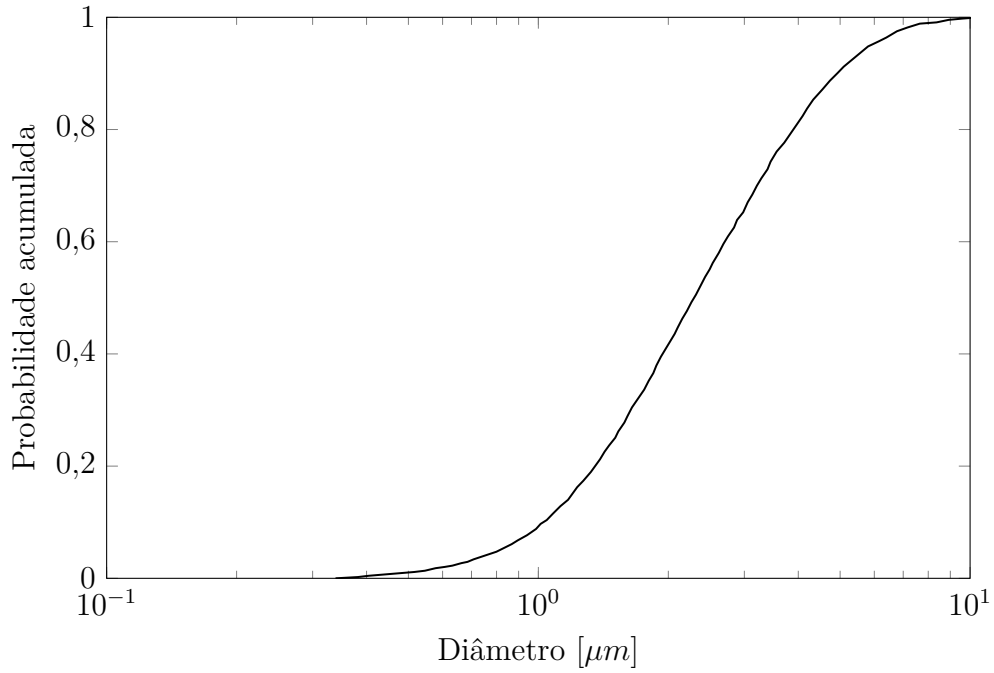


Figura 17 – Distribuição acumulada do tamanho de partículas experimental em massa apresentada por Huang *et al.* (2018)

trabalhos similares como Karimi *et al.* (2012), Sgrott *et al.* (2015) e Balestrin *et al.* (2017). Este procedimento foi descrito por Celik *et al.* (2008) no qual o primeiro passo é definir o tamanho da malha l como,

$$l = \left[\frac{1}{c} \sum_{i=1}^c (\Delta V_i) \right]^{\frac{1}{3}}, \quad (3.77)$$

para geometrias em três dimensões, na qual ΔV_i é o volume de cada célula i e c é o número total de células da malha.

Depois deve-se propor três malhas com refinamentos diferentes de modo que $l_1 < l_2 < l_3$ e selecionar uma variável ϕ de interesse para o estudo. Define-se o fator de refinamento entre duas malhas (R) e a ordem aparente (o), respectivamente, como:

$$R_{21} = \frac{l_2}{l_1}, \quad R_{32} = \frac{l_3}{l_2}, \quad (3.78)$$

$$o = \frac{1}{\ln(R_{21})} \left| \ln \left| \frac{\varepsilon_{32}}{\varepsilon_{21}} \right| + q(o) \right|, \quad (3.79)$$

na qual,

$$\varepsilon_{32} = \phi_3 - \phi_2, \quad \varepsilon_{21} = \phi_2 - \phi_1, \quad (3.80)$$

e

$$q(o) = \ln \left(\frac{R_{21}^o - s}{R_{32}^o - s} \right), \quad (3.81)$$

em que

$$s = 1 \cdot \operatorname{sgn} \left(\frac{\varepsilon_{32}}{\varepsilon_{21}} \right). \quad (3.82)$$

Espera-se que os valores da variável de interesse convirjam para um valor extrapolado formando uma região assintótica. A variável extrapolada é definida como:

$$\phi^{ext} = \frac{R_{21}^o \phi_1 - \phi_2}{R_{21}^o - 1}. \quad (3.83)$$

Ainda deve-se definir o erro relativo da variável e da variável extrapolada, respectivamente, como:

$$E_{21} = \left| \frac{\phi_1 - \phi_2}{\phi_1} \right|, \quad (3.84)$$

$$E^{ext} = \left| \frac{\phi^{ext} - \phi_1}{\phi^{ext}} \right|. \quad (3.85)$$

Assim, finalmente, pode-se escrever o GCI para a malha refinada como:

$$GCI_1 = F \frac{E_{21}}{R_{21}^o - 1}, \quad (3.86)$$

na qual F é um fator de segurança que pode variar entre 1,25 e 3 como sugerido por Roache (1994), neste trabalho o valor usado foi de 1,25.

Deste modo, espera-se a convergência do valor da variável de interesse para uma região assintótica, de modo que o GCI seja menor que 5% uma vez que este representa o erro da malha em relação a uma malha extrapolada de maior refinamento possível.

Assim, a fim de avaliar a incerteza numérica das variáveis queda de pressão, eficiência de coleta e perfil de velocidade associada ao tamanho das malhas foram desenvolvidas três malhas hexaédricas com refinamentos diferentes, uma contendo 237280 células, outra com 438159 e a mais refinada com 765417 células. Tanto a geometria quanto as malhas foram geradas no *software* comercial ICEM CFD 17.2. Deste modo, foram realizadas as simulações multifásicas das diferentes malhas utilizando uma abordagem Euleriana-Euleriana com o diâmetro fixo de $1,9 \mu m$, o qual representa o diâmetro médio de Sauter da distribuição do tamanho de partículas na entrada do ciclone apresentada na Figura 17. Os esquemas e algoritmos numéricos utilizados na simulação estão resumidos na Tabela 10. Foram simulados 60 s de tempo real de modo a atingir estado quasi-estacionário e, deste tempo em diante, foram calculados os perfis fluidodinâmicos médios durante mais 30 s de tempo real.

3.2.6.2 Análise da influencia do número de pontos de quadratura

Após a escolha da malha, foi analisada a influência do número de pontos de quadratura nos perfis fluidodinâmicos de ciclones com alta carga de sólido. Para isso, primeiramente, a partir da distribuição mássica acumulada do tamanho de partículas apresentada na Figura 17 foi calculada a função densidade de probabilidade de número representada na Figura 18, cujo os doze primeiros momentos se encontram na Tabela 11.

Tabela 10 – Esquemas e algoritmos numéricos utilizados na simulação do teste de malha para o ciclone com alta carga

Termo	Modelo
Acoplamento Pressão-Velocidade	<i>Phase Coupled SIMPLE</i>
Gradiente	<i>Green-Gauss Cell Based</i>
Momento	<i>Second Order Upwind</i>
Fração Volumétrica	<i>First Order Upwind</i>
Energia Cinética Turbulenta	<i>First Order Upwind</i>
Taxa de Dissipação Turbulenta	<i>First Order Upwind</i>
Tensão de Reynolds	<i>First Order Upwind</i>
Transiente	<i>First Order Implicit</i>

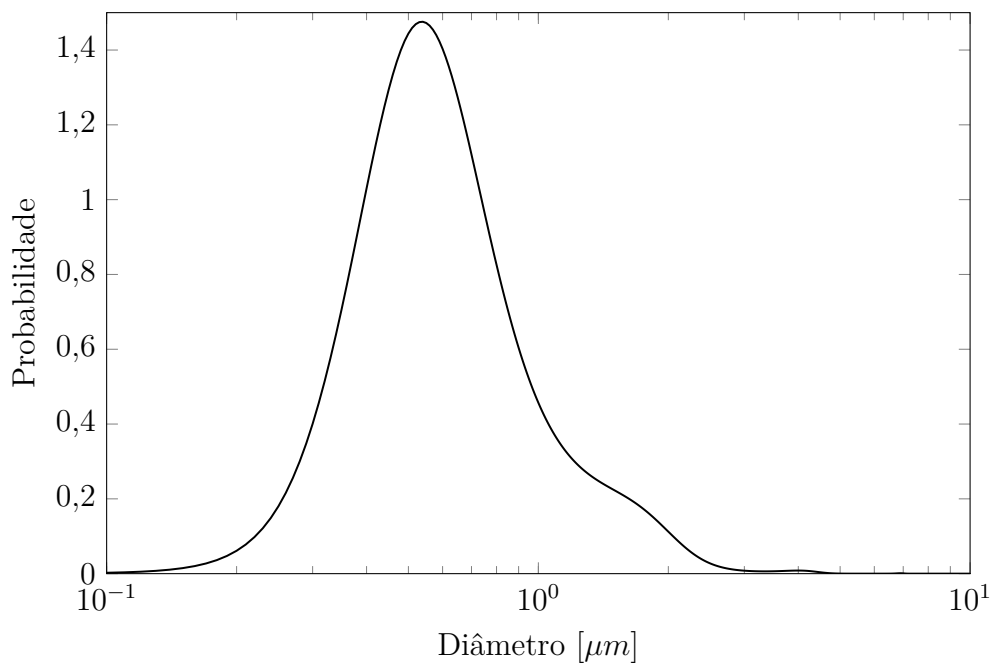


Figura 18 – Função densidade de probabilidade de número utilizada para simulação do ciclone com alta carga de sólidos

De posse dos momentos da distribuição do tamanho de partícula e utilizando o algoritmo Produto-Diferença, cujo o código se encontra disponível no Apêndice C, calculou-se os pesos e abscissas para os casos com dois, três, quatro, cinco e seis pontos de quadratura. E, ainda, empregando a Equação (3.28) foi calculada a fração volumétrica para todos os casos. Esses dados foram sintetizados na Tabela 12.

Sendo assim, para a simulação de ciclones de alta carga utilizando a abordagem proposta neste trabalho foram consideradas as condições de contorno apresentadas na seção 3.2.6 e os mesmos modelos utilizados no teste de independência de malhas, sendo necessário acrescentar as condições de contorno para os pontos de quadratura. Deste modo, na entrada do ciclone foi utilizada a condição de fluxo constante, na parede, fluxo igual a zero, na saída do *vortex finder* foi considerado que a fração volumétrica do centro

Tabela 11 – Momentos da função densidade de probabilidade de número utilizada para simulação do ciclone com alta carga de sólidos

Ordem do momento	Valor
0	1
1	$8,9883 \cdot 10^{-1} \mu m$
2	$1,1594 \mu m^2$
3	$2,2025 \mu m^3$
4	$6,0397 \mu m^4$
5	$2,2706 \cdot 10^1 \mu m^5$
6	$1,0966 \cdot 10^2 \mu m^6$
7	$6,4031 \cdot 10^2 \mu m^7$
8	$4,3066 \cdot 10^3 \mu m^8$
9	$3,2162 \cdot 10^4 \mu m^9$
10	$2,5948 \cdot 10^5 \mu m^{10}$
11	$2,2164 \cdot 10^6 \mu m^{11}$

Tabela 12 – Os pesos (w), as abcissas (ξ) e as frações volumétricas (α) de cada ponto de quadratura (q) para os casos com o total de N pontos de quadraturas da condição de contorno de entrada do ciclone de alta carga

		q					
		1	2	3	4	5	6
$N = 2$	w	0,893	0,107	-	-	-	-
	α	0,135	0,865	-	-	-	-
	$\xi [\mu m]$	0,693	2,61	-	-	-	-
$N = 3$	w	0,771	0,225	0,004	-	-	-
	α	0,075	0,643	0,282	-	-	-
	$\xi [\mu m]$	0,600	1,84	5,35	-	-	-
$N = 4$	w	0,668	0,314	0,018	0,0002	-	-
	α	0,048	0,467	0,432	0,053	-	-
	$\xi [\mu m]$	0,54	1,49	3,71	7,89	-	-
$N = 5$	w	0,579	0,373	0,046	0,002	0,00003	-
	α	0,0335	0,332	0,460	0,160	0,0145	-
	$\xi [\mu m]$	0,504	1,25	2,80	5,53	9,36	-
$N = 6$	w	0,507	0,401	0,085	0,007	0,00036	0,00001
	α	0,025	0,233	0,431	0,248	0,056	0,007
	$\xi [\mu m]$	0,475	1,08	2,24	4,25	6,96	9,99

da célula computacional seria o valor que deixaria o ciclone. Esta condição de contorno precisou ser implementada e o código está disponível no Apêndice B.3. Foi considerado ainda que todo sólido que atingisse o *hopper* seria retirado do sistema. Esta condição de contorno foi implementada junto com a correção da fração volumétrica utilizando a sub-

rotina *DEFINE_EXECUTE_AT_END*. Por fim, para a simulação da fração volumétrica dos pontos de quadratura foi utilizado o esquema de interpolação de primeira ordem. Foram simulados 10 s de tempo real de modo a atingir estado quasi-estacionário e, deste tempo em diante, foram calculados os perfis fluidodinâmicos médios durante mais 10 s de tempo real.

3.2.6.3 Análise da influencia do arraste sólido-sólido

Baseado na literatura, foi proposto utilizar um modelo de arraste de quatro vias, considerando a interação sólido-sólido, para a modelagem de ciclones com alta carga. Contudo, para entender melhor o efeito desta interação no comportamento fluidodinâmico e nos parâmetros de desempenho do ciclone, foi proposto simular o caso com seis pontos de quadratura utilizando um modelo de arraste de duas vias. Deste modo a Equação 3.50 para o cálculo das velocidades dos pontos de quadratura pode ser simplificada a:

$$K_{gq}(\mathbf{v}_g - \mathbf{v}_q) = \frac{\alpha_q}{\alpha_s} K_{gs}(\mathbf{v}_g - \mathbf{v}_s). \quad (3.87)$$

Com isso, foi modificada a sub-rotina do método de Newton para este modelo de duas vias e não foi mais necessário utilizar o método de eliminação de Gauss, uma vez que estas equações não lineares podem ser resolvidas separadamente. Assim, a simulação foi realizada utilizando os mesmos modelos e condições do caso com arraste de quatro vias e utilizando seis pontos de quadratura.

3.2.6.4 Avaliação de outras condições operacionais

Por fim, foi proposto avaliar a utilização desta metodologia para outros casos modificando a velocidade de entrada e a carga de sólidos. A Tabela 13 apresenta, primeiramente, o caso de referência que foi utilizado nas seções anteriores e os novos casos propostos.

Tabela 13 – Velocidade, carga de sólidos e fração volumétrica da entrada do ciclone

Casos	Velocidade de entrada [$m \cdot s^{-1}$]	Carga de sólidos [$g \cdot m^{-3}$]	Fração volumétrica de sólidos
1	15	115,3	$3,98 \cdot 10^{-5}$
2	15	57,7	$1,99 \cdot 10^{-5}$
3	18	96,1	$3,314 \cdot 10^{-5}$
4	18	48,1	$1,658 \cdot 10^{-5}$

Neste estudo foram utilizados as condições e modelos apresentados na seção 3.2.6.2 para o caso com seis pontos de quadratura.

4 RESULTADOS

Este capítulo apresenta os principais resultados obtidos durante o desenvolvimento desta dissertação. Os resultados estão subdivididos na implementação e verificação das sub-rotinas, avaliação do número de pontos de quadratura para leito fluidizado circulante e, por fim, os resultados obtidos para a simulação de ciclones com alta carga de sólidos.

4.1 IMPLEMENTAÇÃO E VERIFICAÇÃO DAS SUB-ROTINAS

O código implementado para acoplamento da PBE à abordagem Euleriana-Euleriana contida no Fluent está descrito no Apêndice A. A Figura 19 apresenta, a título de exemplo, a sub-rotina *DEFINE_UDS_FLUX* implementada para definir o termo convectivo das equações de transporte resultantes da aplicação do DQMOM para a solução da PBE.

```

/*----- Define o termo convectivo -----*/

DEFINE_UDS_FLUX(my_uds_flux_PBE,f,t,i)
{
    cell_t c0, c1 = -1;
    Thread *t0, *t1 = NULL;
    real NV_VEC(psi_vec), NV_VEC(A), flux = 0.0;
    c0 = F_C0(f,t);
    t0 = F_C0_THREAD(f,t);
    F_AREA(A, f, t);
    /* Se pertencer a uma condição de contorno, usar valores da face; */
    /* caso contrário, usar média das células adjacentes. */
    if (BOUNDARY_FACE_THREAD_P(t))
    {
        NV_DS(psi_vec, =, F_UDMI(f,t,3*i), F_UDMI(f,t,3*i+1), F_UDMI(f,t,3*i+2), *,rho_s);
        flux = NV_DOT(psi_vec, A); /* fluxo atraves da face */
    }
    else
    {
        c1 = F_C1(f,t); /* célula vizinha a face */
        t1 = F_C1_THREAD(f,t);
        NV_DS(psi_vec, =, C_UDMI(c0,t0,3*i), C_UDMI(c0,t0,3*i+1), C_UDMI(c0,t0,3*i+2), *,rho_s);
        NV_DS(psi_vec, +=, C_UDMI(c1,t1,3*i), C_UDMI(c1,t1,3*i+1), C_UDMI(c1,t1,3*i+2), *,rho_s);
        flux = NV_DOT(psi_vec, A)/2.0; /* fluxo médio através da face */
    }
}

```

Figura 19 – Sub-rotina para definir o termo convectivo das equações de transporte resultantes da aplicação do DQMOM na PBE

É importante ressaltar que uma das dificuldades foi fazer a comunicação entre o código escrito e o *software* Fluent. Assim, foi necessário ter conhecimento de alguns conceitos, como por exemplo o conceito de domínio que representa um conjunto de todas as células e faces da malha contendo as variáveis da mistura ou de cada fase e o conceito de *thread*, que está contido no domínio e representa um conjunto de células ou faces que apresentam uma característica semelhante, como por exemplo, uma condição de contorno, além de funções específicas para ter acesso aos valores das variáveis das células ou faces obtidas pelo Fluent. A Tabela 14 apresenta algumas destas funções.

Tabela 14 – Funções utilizadas para obter o valor de variáveis em cada célula da malha

Variável	Função
Fração volumétrica	C_VOF(célula, <i>thread</i>)
Velocidade em x	C_U(célula, <i>thread</i>)
Velocidade em y	C_V(célula, <i>thread</i>)
Velocidade em z	C_W(célula, <i>thread</i>)
Volume	C_VOLUME(célula, <i>thread</i>)
i-ésimo escalar	C_UDSI(célula, <i>thread</i> , i)
i-ésima memória	C_UDMI(célula, <i>thread</i> , i)

No intuito de verificar as sub-rotinas implementadas, foram obtidos os resultados para a simulação do leito fluidizado circulante, com base na metodologia proposta na seção 3.2.2, e comparados com os obtidos por Niemi (2012). A Figura 20 apresenta os perfis de fração volumétrica média da fase sólida desta simulação e os obtidos por Niemi (2012) para as alturas de 23 *cm* (a), 40 *cm* (b), 80 *cm* (c) e 120 *cm* (d).

A partir da análise dos perfis apresentados da Figura 20, pode-se dizer que para alturas mais elevadas do CFB como nas Figuras 20 (c) e (d), as quais possuem uma menor fração volumétrica, a simulação obteve resultados muito similares aos atingidos por Niemi (2012) e bem próximos aos experimentais. Já para as alturas inferiores como nas Figuras (a) e (b), as quais possuem uma maior fração volumétrica, pode-se observar uma diferença entre os resultados da simulação e dos dados experimentais. Niemi (2012) sugere que uma possível explicação está na dificuldade de se obter dados experimentais utilizando velocimetria por imagem de partículas (*particle image velocimetry* - PIV) em regiões com alta fração volumétrica de sólidos. De acordo com Peltola *et al.* (2010), os valores de fração volumétrica da partícula são estimados a partir da intensidade da luz em imagem de PIV e são, normalmente, superestimados em regiões com maiores concentrações sólidos.

Além da fração volumétrica, outra variável importante a se avaliar são as velocidades ao longo do leito. Sendo assim, a Figura 21 apresenta o perfil da velocidade vertical média da fase sólida nas alturas de 23 *cm* (a), 40 *cm* (b), 80 *cm* (c) e 120 *cm* (d). Pode-se perceber pela análise da Figura 21 que há uma boa correspondência entre a simulação e os dados experimentais especialmente para região de escoamento desenvolvido

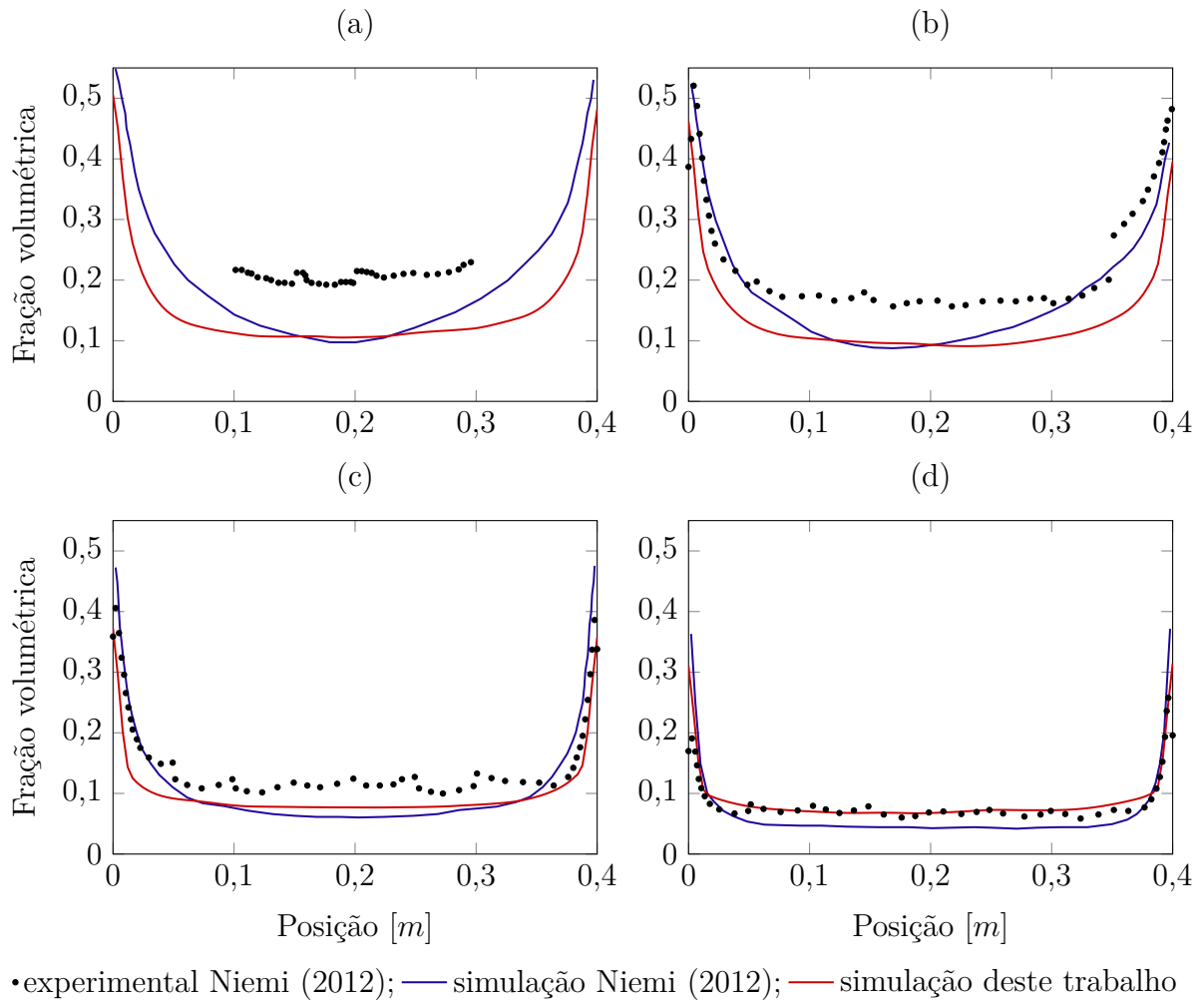


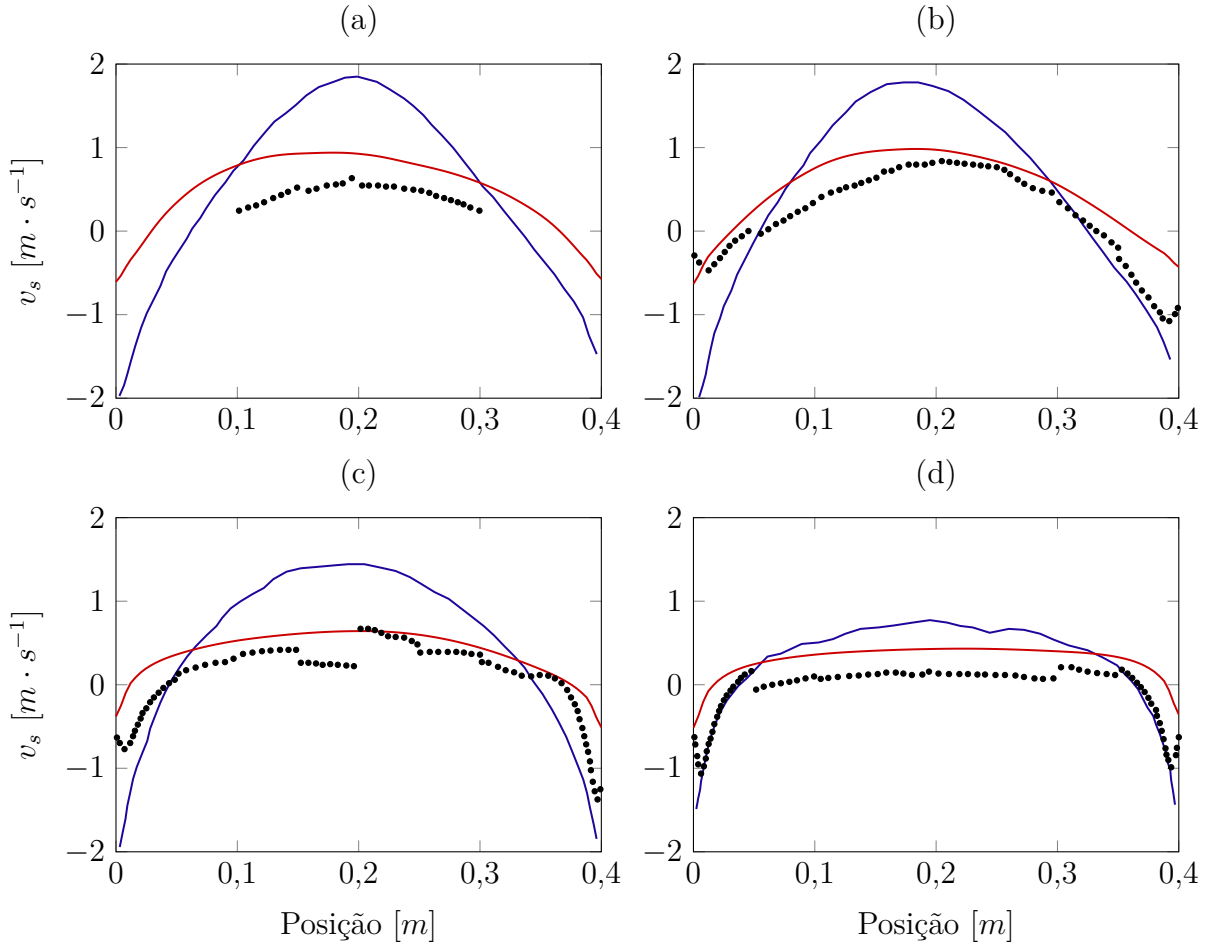
Figura 20 – Fração volumétrica média da fase sólida em diferentes alturas do leito: (a) 23 cm; (b) 40 cm; (c) 80 cm e (d) 120 cm

(Figura 21 (c) e (d)).

Em contrapartida, os resultados obtidos pela simulação de Niemi (2012) não se ajustaram muito bem aos experimentais. Isso se deve ao fato de que foi utilizada uma média de Frave ponderada pela fração volumétrica de sólidos para calcular o perfil da velocidade vertical médio mostrado na Figura 21, sendo esta não utilizada neste trabalho.

Em adição, a Figura 22 apresenta a média na seção transversal do diâmetro médio de Sauter como uma função da altura do leito. Observa-se que os resultados obtidos através desta simulação foram próximos aos obtidos por Niemi (2012), diferindo apenas em alturas mais elevadas do leito. Este comportamento pode ser explicado devido ao fato de não se saber qual a condição de contorno utilizada por Niemi (2012) para os pesos das quadraturas na saída do leito fluidizado circulante.

Fica evidenciado ainda pela Figura 22, que utilizando dois pontos de quadratura foi possível observar uma segregação das partículas no CFB, de modo que, acima de



•experimental Niemi (2012); — simulação Niemi (2012); — simulação deste trabalho

Figura 21 – Perfil da velocidade vertical média em diferentes alturas do leito: (a) 23 cm; (b) 40 cm; (c) 80 cm e (d) 120 cm

1,5 m o SMD diminui em comparação ao SMD utilizado na simulação monodispersa indo para próximo do valor da menor abscissa, ficando clara a importunância e a influência da simulação polidispersa nos perfis fluidodinâmicos.

De modo a complementar a influência do modelo polidisperso nos perfis fluidodinâmicos observada para SMD, avaliou-se os resultados para as mapas da fração volumétrica média. A Figura 23 apresenta os mapas para o primeiro (a) e segundo (b) ponto de quadratura, da fase sólida total considerando a soma do valores das duas quadraturas (c) e o perfil do modelo monodisperso utilizando apenas o diâmetro médio de Sauter inicial de $279 \mu m$ (d). Dessa forma, pode-se dizer que a quadratura de maior abscissa, mostrada na Figura 23 (b), apresentou uma maior concentração de partículas na parte inferior do leito comparada a de menor abscissa, o que era esperado uma vez que partículas de menor diâmetro são carreadas mais facilmente.

No entanto, apesar das diferenças entres os dois pontos de quadraturas, o mapa da fração volumétrica média da fase sólida total, apresentado na Figura 23 (c) não diferiu

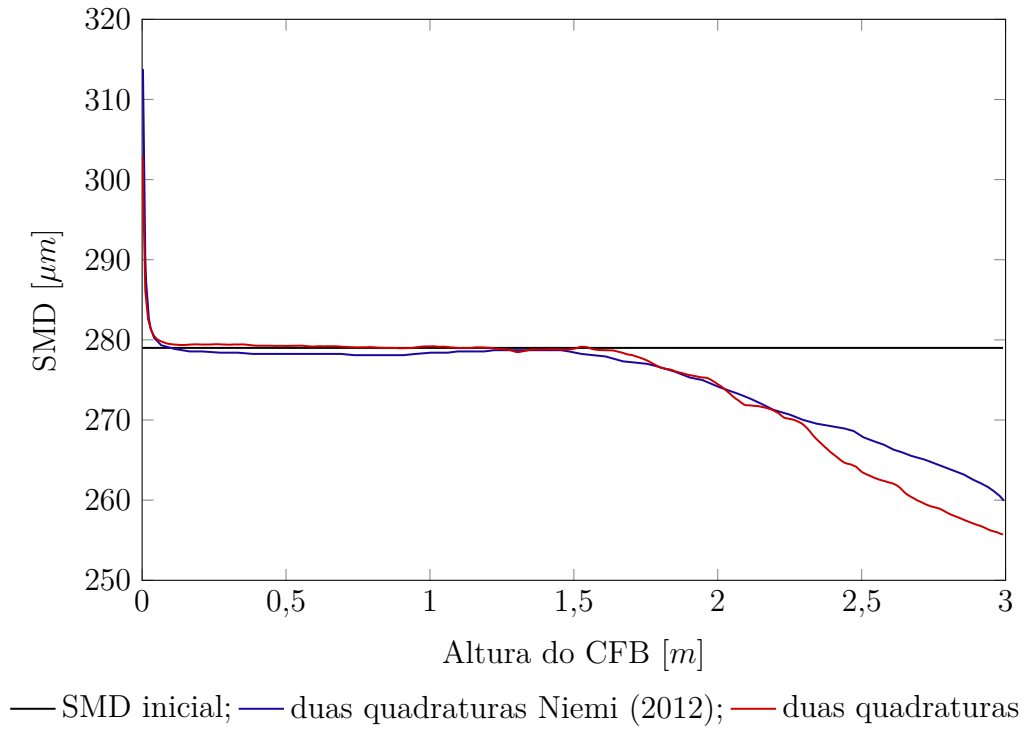


Figura 22 – Média transversal do diâmetro médio de Sauter em função da altura do CFB

significativamente da monodispersa, Figura 23 (d). Foi possível perceber que na parte superior do leito há uma maior concentração de sólidos na simulação com dois pontos de quadratura.

A partir do que foi observado pode-se dizer que apesar dos resultados desta simulação possuírem algumas divergências dos obtidos por Niemi (2012), eles se mostram mais próximos aos dados experimentais. Sendo assim, pode-se considerar que o código foi verificado.

De maneira a estender o estudo de Niemi (2012), faz-se necessário analisar o número de quadraturas necessário para se obter uma boa aproximação da distribuição do tamanho de partículas e para se obter simulações com resultados similares dentro de um erro determinado.

4.2 AVALIAÇÃO DO NÚMERO DE QUADRATURAS

Os resultados desta seção foram subdivididos nos dois testes propostos: reconstrução da função densidade de probabilidade e na simulação do leito fluidizado circulante.

4.2.1 Reconstrução da Função Densidade de Probabilidade

Foi reconstruída a distribuição do tamanho de partículas inicial empregando os métodos de máxima entropia e *spline* cúbica, no intuito de avaliar a influência o número de

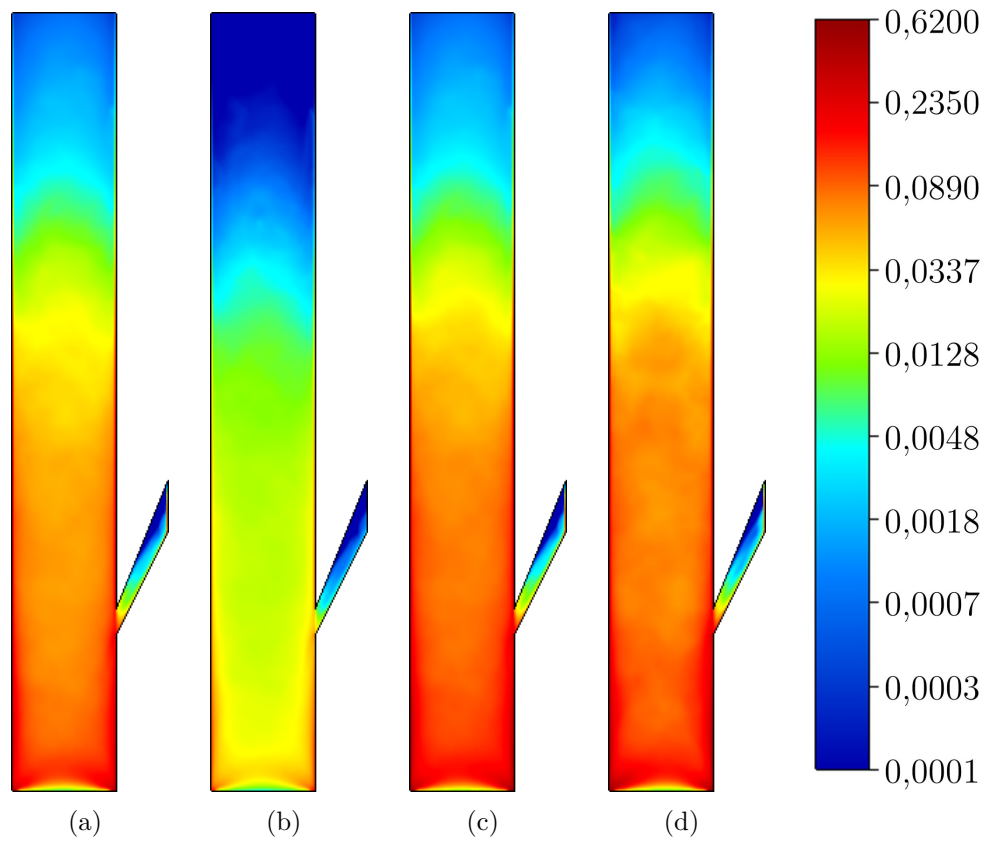


Figura 23 – Mapas da fração volumétrica média no leito fluidizado circulante. (a) ponto de quadratura com abscissa de $255 \mu m$; (b) ponto de quadratura com abscissa de $416 \mu m$; (c) fase sólida total considerando a soma dos dois pontos de quadratura e (d) monodispersa utilizando o diâmetro médio de Sauter inicial de $279 \mu m$

momentos no erro da reconstrução da curva. Assim, a Figura 24 apresenta as reconstruções da PSD inicial empregando os métodos de máxima entropia e *spline* cúbica utilizando três, quatro, cinco, seis, sete, oito, nove ou dez momentos, os quais estão presentes da Tabela 8. Os erros relativos¹ dessas reconstruções estão representados na Tabela 15.

Pode-se observar pela Figura 24 e Tabela 15 que o método de máxima entropia para os diferentes números de momentos se aproximou mais da curva original do que o método *spline* cúbica, sendo que para 7 momentos o erro relativo foi menor que 3%. O grande número de parâmetros que necessitam ser ajustados para obter convergência no método *spline* cúbica podem ter ocasionado esse maior erro. Sendo assim, as reconstruções oriundas deste método podem sofrer alterações dependendo dos parâmetros utilizados como, intervalo inicial que contem a distribuição, permitir números negativos, menor intervalo permitido entre nós e expansão do domínio.

Deste modo, o método de máxima entropia necessita de um menor número

¹ O erro relativo foi definido como $Erro = \frac{|f(x) - f_R(x)|}{f(x)}$, na qual $f(x)$ é a distribuição inicial e $f_R(x)$ é a distribuição reconstruída

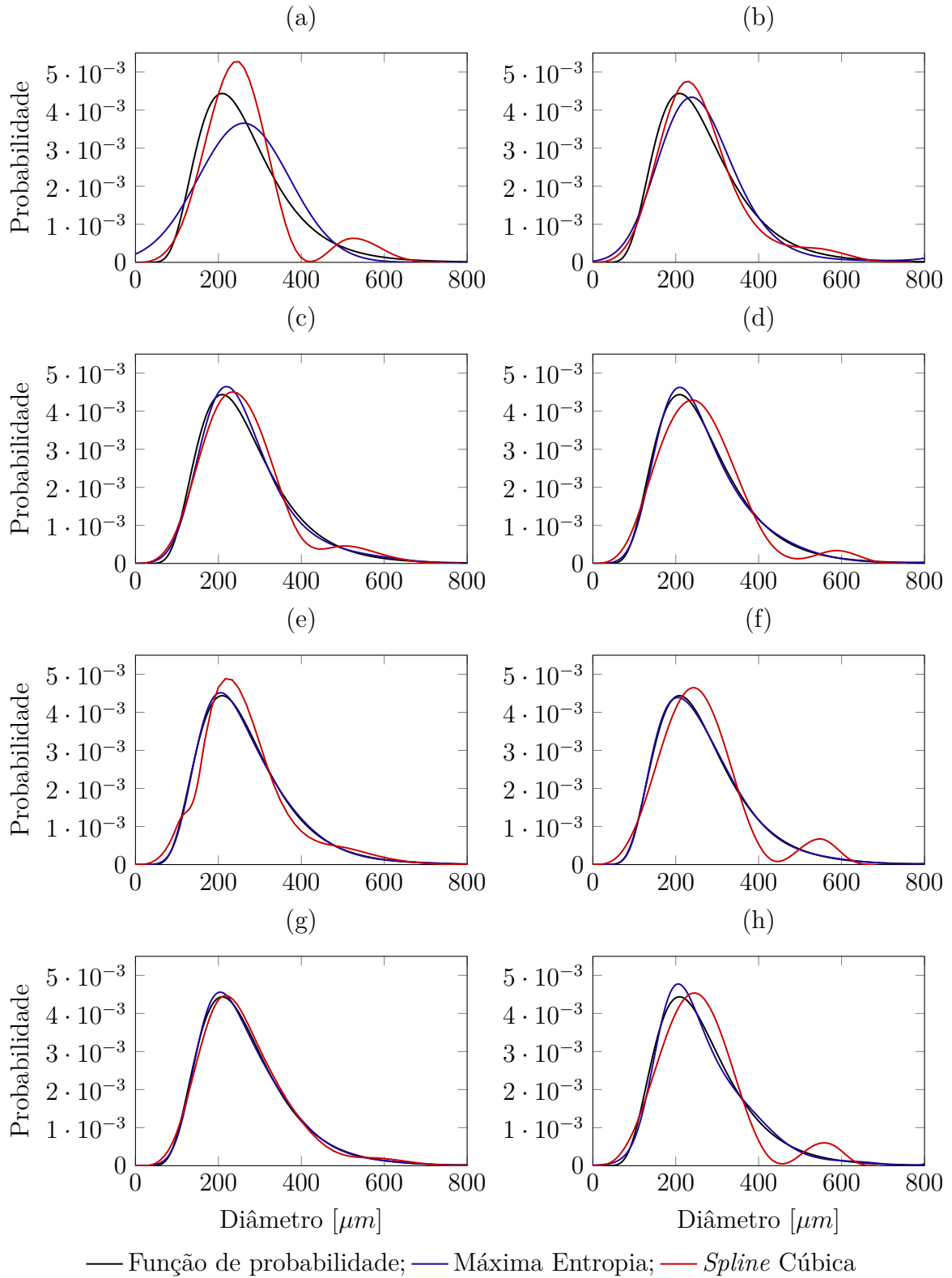


Figura 24 – Reconstrução da distribuição do tamanho de partículas inicial utilizando os métodos máxima entropia e *spline* cúbica e diferentes números de momentos. (a) 3 momentos; (b) 4 momentos; (c) 5 momentos; (d) 6 momentos; (e) 7 momentos; (f) 8 momentos; (g) 9 momentos e (h) 10 momentos

Tabela 15 – Erro relativo das reconstruções utilizando os métodos de máxima entropia e *spline* cúbica

Momentos	Erro relativo [%]	
	Máxima entropia	<i>Spline</i> cúbica
3	28,84	57,21
4	18,59	20,79
5	8,17	26,90
6	4,24	30,15
7	2,70	22,94
8	2,31	53,18
9	2,78	9,49
10	7,32	51,89

de parâmetros de entrada e apresentou um erro menor que 3% utilizando pelo menos 7 momentos, ou seja, são necessários no mínimo quatro pontos de quadratura. Importante ressaltar ainda que o aumento do erro apresentado por esse método quando utilizado 9 e 10 momentos foi devido geração de uma matriz singular durante a solução do problema de otimização.

4.2.2 Simulação do Leito Fluidizado Circulante

No intuito de avaliar o efeito do número de quadraturas sobre o comportamento fluidodinâmico do leito fluidizado, comparou-se o resultado das simulações utilizando 2, 3, 4, 5 e 6 pontos de quadratura além da simulação monodispersa. A Figura 25 apresenta os mapas para as frações volumétricas para os casos estudados. Percebe-se que a utilização dessa metodologia obteve uma maior fração volumétrica de sólidos no topo do CFB comparado com o resultado obtido pela simulação monodispersa. Além disso, pode-se dizer pela Figura 25 que qualitativamente o mapa da fração volumétrica média converge, sendo que o mapa para 4 pontos de quadratura se assemelha aos perfis de 5 e 6 pontos.

Assim sendo, para avaliar quantitativamente o efeito do número de quadraturas, a Figura 26 apresenta a média na seção transversal dos momentos de ordem 1 (a), 2 (b), 3 (c) e 4 (d) em função da altura do leito. A Tabela 16 apresenta o erro relativo² entre as quadraturas dos momentos calculados. Percebe-se que há convergência a medida que aumenta o número de pontos de quadraturas sendo que o erro relativo entre 4 pontos e 5 pontos é menor que 10% para os momentos de ordem 1 a 4. Deste modo, novamente, a partir de 4 pontos de quadratura tem-se uma aproximação aceitável do perfil fluidodinâmico de um sistema polidisperso.

² O erro relativo para o momento de ordem k foi definido como $Erro_i = \frac{|M_k^i(x) - M_k^{i+1}(x)|}{M_k^{i+1}(x)}$, na qual $M_k^i(x)$ é o momento de ordem k da quadratura i e $M_k^{i+1}(x)$ é o momento de ordem k da quadratura $i + 1$ em função da altura x

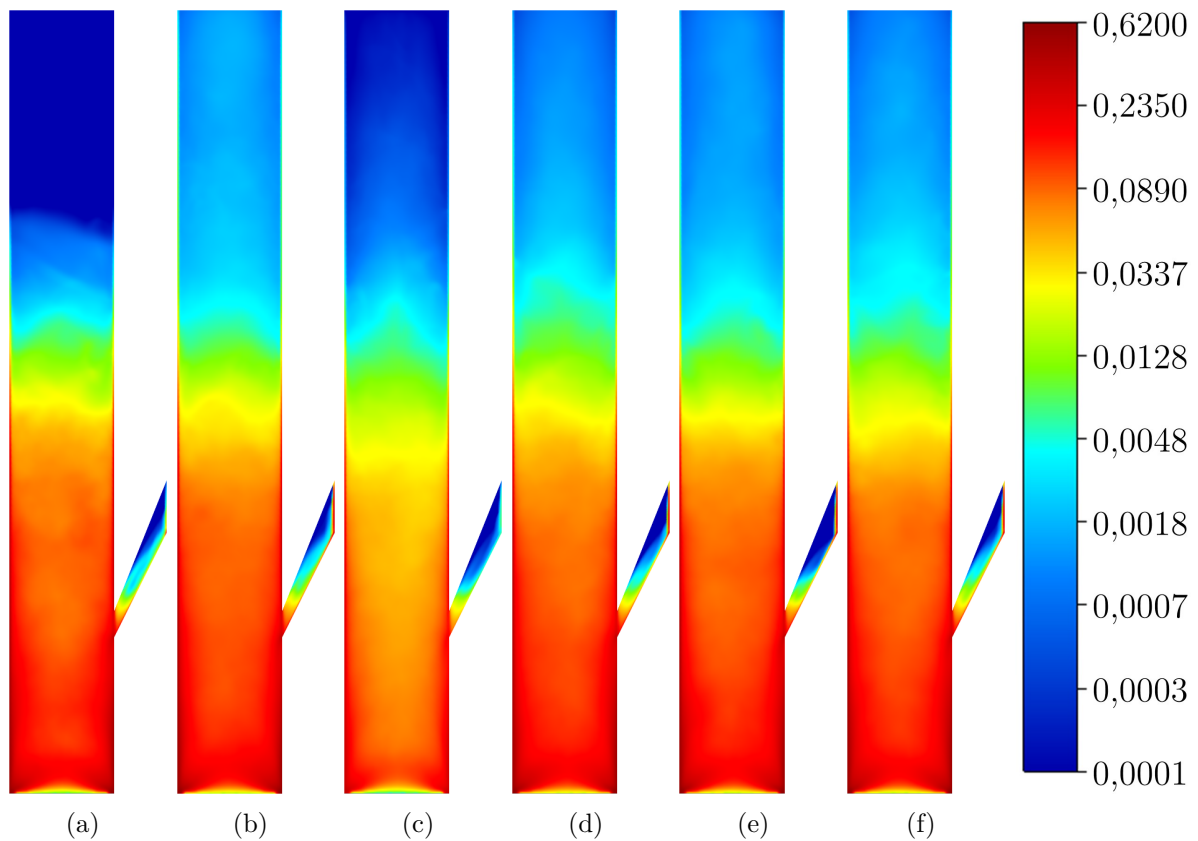


Figura 25 – Comparação dos mapas da fração volumétrica média no CFB para a simulação monodispersa e polidispersa com diferentes pontos de quadratura. (a) monodispersa; (b) dois pontos de quadratura; (c) três pontos de quadratura; (d) quatro pontos de quadratura; (e) cinco pontos de quadratura e (f) seis pontos de quadratura

Tabela 16 – Erro relativo entre as quadraturas para os momentos de ordem 1, 2, 3 e 4

Pontos de quadraturas	Erro relativo [%]			
	Momentos de ordem			
	1	2	3	4
2/3	5,38	9,50	11,88	13,53
3/4	2,89	6,78	10,43	12,88
4/5	1,82	3,55	5,32	7,73
5/6	1,05	2,15	3,66	6,00

Além disso, a Tabela 17 apresenta o tempo das simulações. A partir dela, pode-se observar que o tempo da simulação com 4 pontos de quadratura foi duas vezes menor que o de 6 pontos, sendo possível soluções com erros aceitáveis com custo computacional reduzido.

A fim de acompanhar a variação do diâmetro e avaliar a reconstrução de curvas ao longo do leito fluidizado foram calculados o diâmetro médio de Sauter, desvio padrão, obliquidade e curtose, os quais se encontram na Figura 27. Observando o diâmetro médio

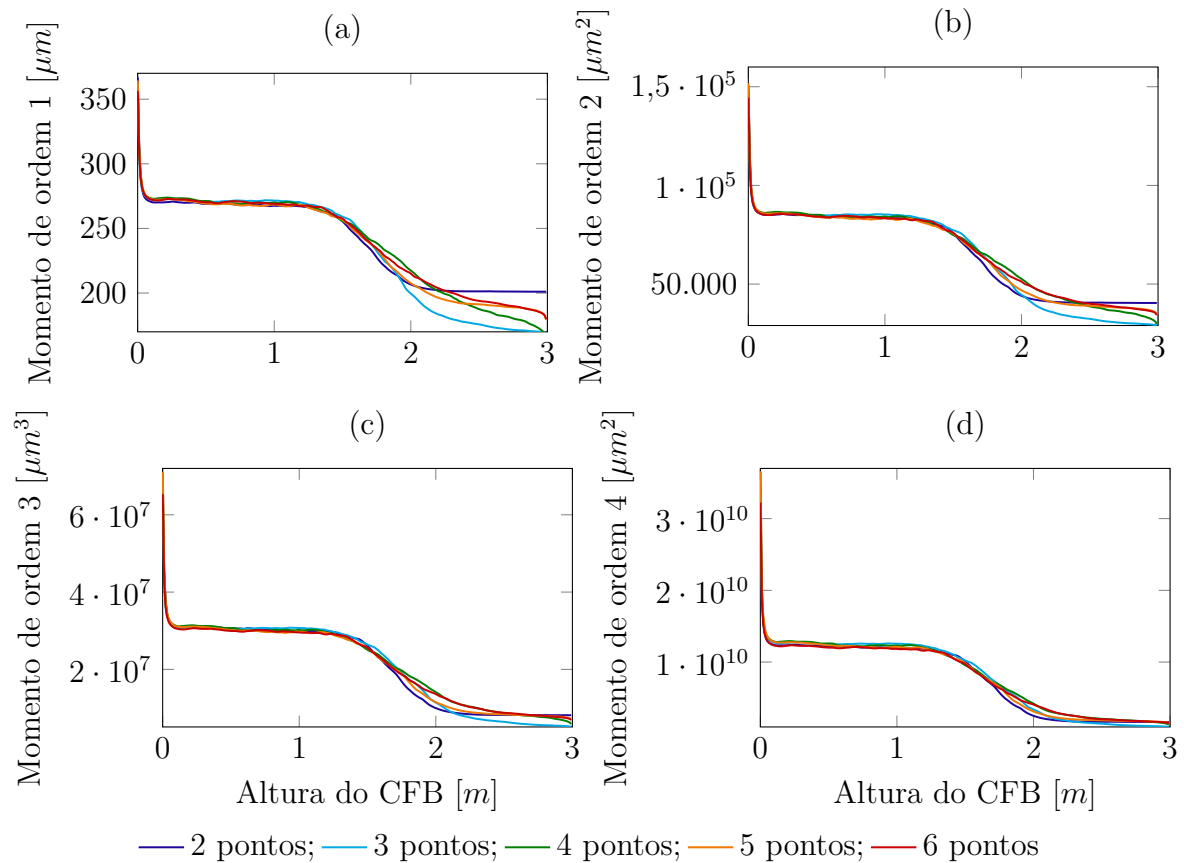


Figura 26 – Média transversal dos momentos da função densidade de probabilidade de número ao longo do leito fluidizado circulante para os diferentes pontos de quadraturas. (a) momento de ordem 1; (b) momento de ordem 2; (c) momento de ordem 3 e (d) momento de ordem 4

Tabela 17 – Tempo de simulação para os diferentes números de pontos de quadratura

Pontos de quadraturas	Tempo de simulação [h]
2	161,8
3	239,3
4	367,7
5	585,0
6	837,2

de Sauter na Figura 27 (a) pode-se salientar novamente que as curvas de 4, 5 e 6 pontos de quadratura se assemelham e ainda pode-se ressaltar a segregação das partículas na qual o fundo do CFB apresenta um diâmetro maior que o topo.

Em relação a reconstrução de curvas foi avaliado as medidas de dispersão, assimetria e forma da PDF. O desvio padrão é uma medida de dispersão da função densidade de probabilidade, ou seja, mede o quanto os dados da distribuição se dispersam da média. Pela Figura 27 (b) é possível observar convergência das curvas de desvio padrão ao longo do leito, sendo que o perfil para quatro pontos de quadratura se assemelha aos

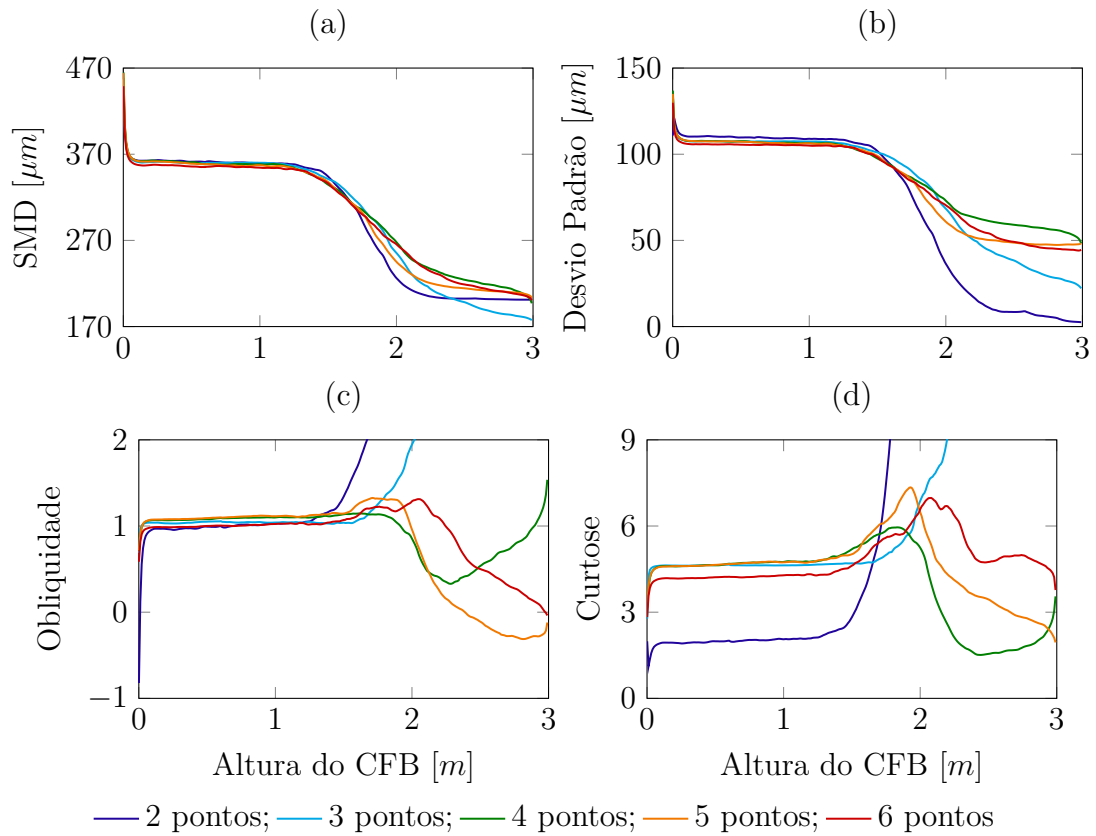


Figura 27 – Média transversal dos diâmetro médio de Sauter e medidas de dispersão, assimetria e forma da função densidade de probabilidade de número ao longo do CFB para diferentes pontos de quadratura. (a) diâmetro médio de Sauter; (b) desvio padrão; (c) obliquidade e (d) curtose

de cinco e seis pontos. Além disso, percebe-se que o desvio padrão diminuiu a medida que aumenta a altura do leito, de modo que o valor na saída é aproximadamente a metade do da entrada para quatro, cinco e seis pontos de quadratura, indicando uma distribuição mais estrita na saída. Vale ressaltar ainda que o desvio padrão para dois pontos de quadratura na saída é próximo de zero mostrando que a distribuição está concentrada na média sendo um pico muito estreito.

A obliquidade é a medida de assimetria da PDF, distribuições simétricas tem obliquidade igual a zero. Deste modo como os valores da obliquidade são maiores que zero significa que a distribuição tem cauda para direita. No entanto, pela Figura 27 (c), percebe-se que para 5 e 6 pontos de quadratura na saída do leito a obliquidade é menor que zero o que implica numa cauda para esquerda. Uma vez que espera-se que partículas de diâmetros maiores não atinjam o topo do leito, a cauda da direita da PSD inicial tem a tendência de encolher e, deste modo, é esperada a diminuição da obliquidade. Sendo assim, o resultado obtido para 4 pontos de quadratura, no qual há um aumento da obliquidade para alturas maiores que 2 m, não é tão condizente com o esperado.

A curtose é a medida de forma da PDF sendo que valores maiores de curtose

implica em um pico mais pontiagudo e valores menores levam ao achatamento do pico tendendo a uma distribuição uniforme na qual possui uma curtose de aproximadamente 1,8. Desta forma, observando a Figura 27 (d) pode-se dizer que como 2 e 3 pontos de quadraturas possuem curtoses muito altas na saída do leito espera-se que a PDF seja um pico muito pontiagudo e que a ordem decrescente esperada da altura dos outros pontos de quadratura seja 6, 4 e 5.

Deste modo, caso o objetivo do trabalho seja avaliar o tamanho médio ou o desvio padrão, quatro pontos de quadratura são suficientes para representá-los. No entanto, se for importante analisar as medidas de assimetria e forma da função densidade de probabilidade, um número maior de pontos de quadratura é necessário.

Sendo assim, observando a Figura 28, que apresenta a reconstrução da função densidade de probabilidade de número na saída do leito pode-se notar exatamente as características ressaltadas pela avaliação das medidas de dispersão, assimetria e forma. Como previsto pelo desvio padrão e pela curtose a reconstrução da distribuição para 2 e 3 pontos de quadraturas são picos pontiagudos com pouca dispersão da média. Pode-se constatar ainda que a obliquidade de 4 pontos de quadratura é positiva enquanto que para 5 e 6 pontos é negativa tendo cauda para esquerda. Por fim, pode-se observar que a ordem das alturas dos picos seguem o esperado pelos os valores da curtose para 4, 5 e 6 pontos de quadratura.

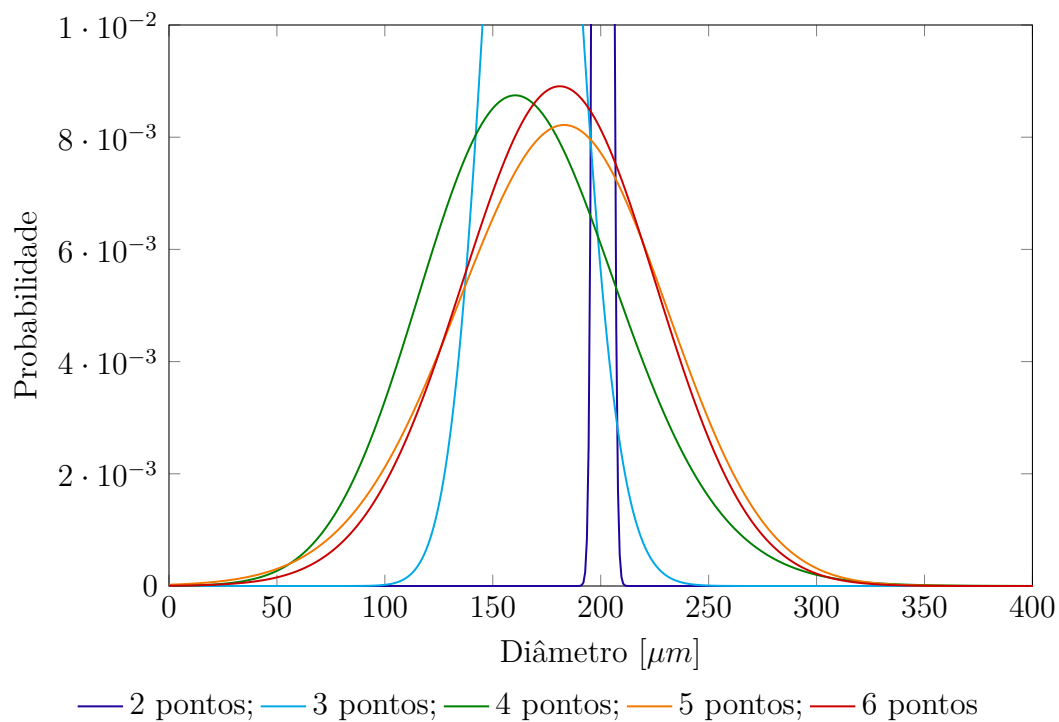


Figura 28 – Reconstrução da função densidade de probabilidade de número na saída do CFB para os diferentes pontos de quadratura

4.3 CICLONE COM ALTA CARGA

Os resultados para as simulações de ciclone com alta carga estão divididos em teste de independência de malha, análise da influência do número de pontos de quadratura, análise da influência do modelo de arraste de quatro vias considerando o arraste sólido-sólido e avaliação de outras condições operacionais.

4.3.1 Teste de Independência de Malhas

Testes de malha foram realizados a luz da metodologia apresentada na seção 3.2.6.1 para avaliar a incerteza numérica das variáveis queda de pressão, eficiência de coleta e velocidade tangencial da fase gasosa. Os resultados da simulação monodispersa para a queda de pressão das malhas grosseira, intermediária e refinada foram, respectivamente, 766,94, 485,20 e 475,92 Pa . Esses resultados estão representados na Figura 29, na qual percebe-se que os valores se encontram numa região assintótica, de modo que um maior refinamento da malha provocaria uma pequena variação dessa variável, sendo o GCI, calculado pela Equação (3.86), para a malha refinada de 0,11%.

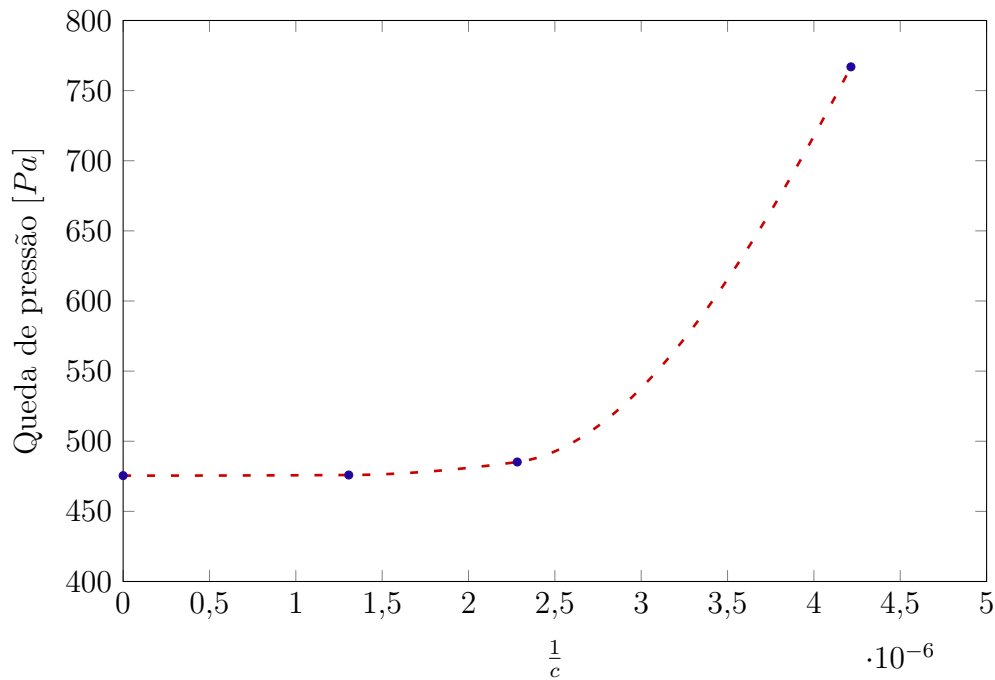


Figura 29 – Queda de pressão pelo inverso do números de células das malhas analisadas e para a extrapolação do método GCI

Do mesmo modo, foram calculadas as eficiências de coleta de sólidos para os diferentes níveis de refinamento das malhas. Os resultados obtidos para ordem crescente de refinamento foram 95,69%, 60,16% e 56,87%, os quais estão representados na Figura 30. Novamente, pela Figura 30, nota-se que os resultados se encontram numa região

assintótica e que o GCI para a malha refinada é de 0,96% salientando que o valor da eficiência global teria uma variação pequena com o refinamento da malha.

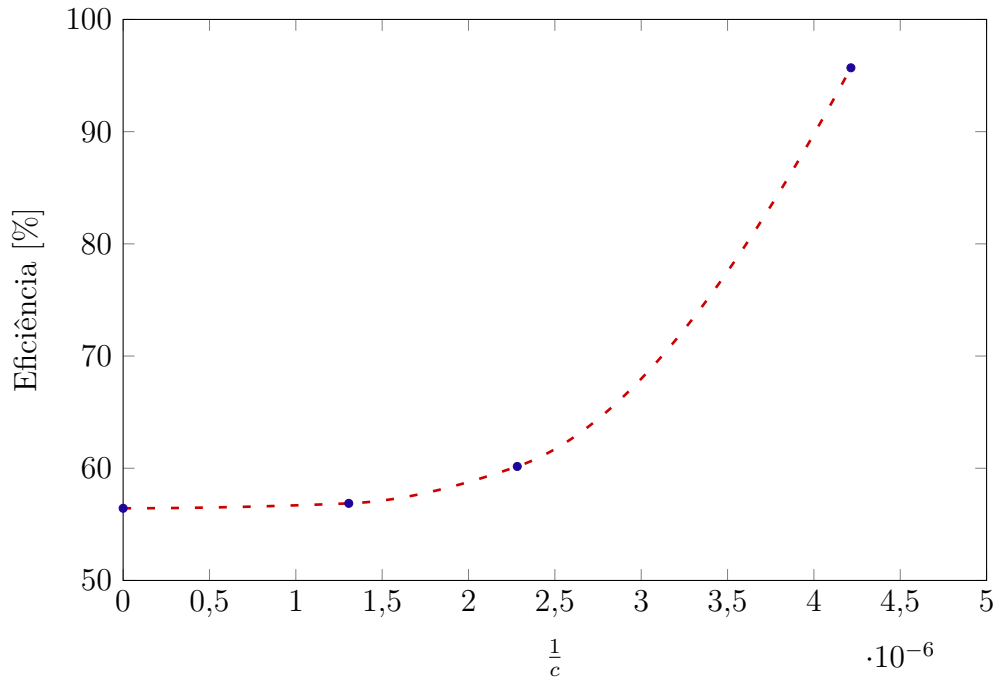


Figura 30 – Eficiência de coleta de sólidos pelo inverso do números de células das diferentes malhas analisadas e para a extrapolação do método GCI

Os resultados para análise de incerteza para variável velocidade tangencial ao longo da posição radial na altura de 216 *mm* são apresentados na Figura 31. Percebe-se através desta que a região de incerteza é pequena mostrando que os resultados obtidos pela malha refinada apresenta um índice de incerteza (GCI) menor que 3% ao longo do raio do ciclone.

Sendo assim, a malha refinada contendo 765417 células foi utilizada para as demais simulações de ciclone.

4.3.2 Análise da Influência do Número de Pontos de Quadratura

No intuito de avaliar a influência do número de pontos de quadratura sobre o comportamento fluidodinâmico do ciclone e também nos seus parâmetros de desempenho, foram realizadas simulações utilizando um modelo monofásico, monodisperso e polidisperso com dois, três, quatro, cinco e seis pontos de quadratura. A simulação utilizando o modelo monofásico foi realizada com o objetivo de comparar o seu resultado da queda de pressão com o obtido experimentalmente por Huang *et al.* (2018). Já para as demais simulação foram comparados os dois parâmetros de desempenho de modo a discutir o efeito dos pontos de quadratura. Assim, os resultados da queda de pressão e eficiência

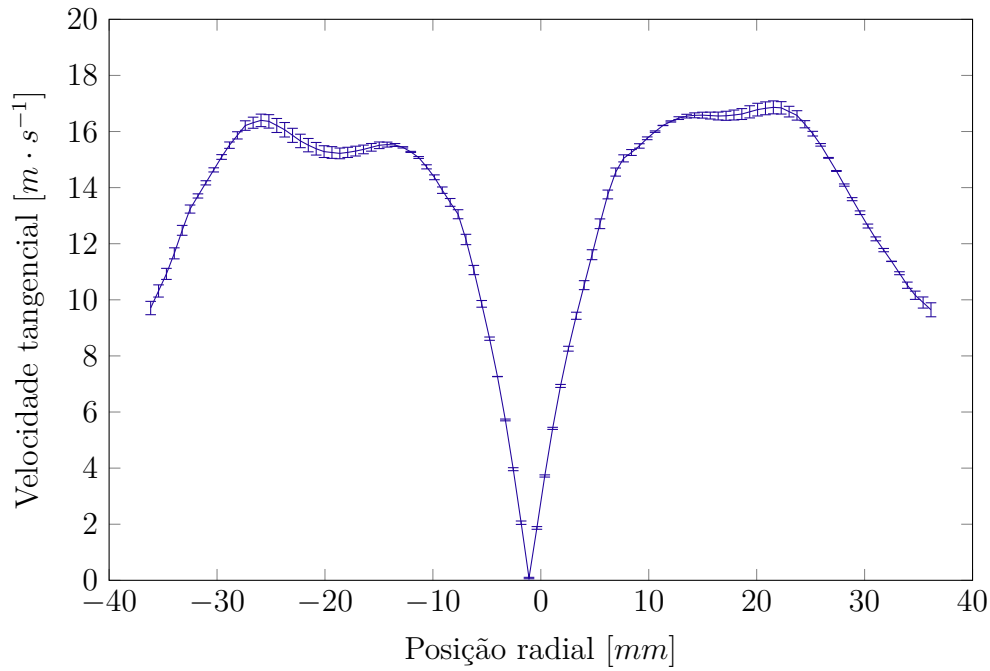


Figura 31 – Incerteza numérica do perfil da velocidade tangencial média da malha refinada calculada utilizando método do GCI na altura de 216 mm logo abaixo da entrada do *vortex finder*

global obtidos pelas as simulações e experimentalmente por Huang *et al.* (2018) estão resumidos na Tabela 18.

Tabela 18 – Comparação da queda de pressão e eficiência global entre diferentes simulações

Experimental/Simulação	ΔP [Pa]	η [%]
experimental monofásico Huang <i>et al.</i> (2018)	600	-
monofásica	662,3	-
experimental Huang <i>et al.</i> (2018)	-	84,2
monodispersa	475,9	56,9
2 pontos de quadratura	697,2	83,3
3 pontos de quadratura	701,7	84,8
4 pontos de quadratura	702,8	87,6
5 pontos de quadratura	706,4	86,5
6 pontos de quadratura	711,5	84,4

Pela análise da Tabela 18, percebe-se que a simulação monofásica apresentou uma queda de pressão um pouco maior que a obtida experimentalmente por Huang *et al.* (2018), no entanto como não se sabe como este resultado foi obtido experimentalmente, dependendo do local da medição da pressão no *vortex finder* pode-se obter valores diferentes devido a pressão dinâmica causada pelo escoamento em espiral. Além disso, pode-se observar que o valor da eficiência global obtido pelas simulações utilizando um modelo polidisperso muito próximo ao obtido experimentalmente, sendo que a simulação com seis

pontos de quadratura obteve um erro relativo³ menor que 1%. Além do mais, pela análise da Tabela 18, nota-se que a simulação utilizando o modelo monodisperso apresentou resultados para queda de pressão e eficiência de coleta bem menores que os obtidos pela simulação polidispersa e experimentalmente, demonstrando assim, a possibilidade do uso da metodologia proposta em detrimento da Euleriana-Euleriana tradicional.

De modo a comparar qualitativamente o efeito do número de pontos de quadratura sobre a fração volumétrica de sólidos, foram criados mapas da fração volumétrica média de sólidos das simulações monodispersa e polidispersa com dois, três, quatro, cinco e seis pontos de quadratura apresentados na Figura 32. Por esta comparação percebe-se que os mapas das simulações polidispersas são bem similares, o mesmo não pode ser observado para a simulação monodispersa que apresentou uma fração volumétrica de sólidos maior (maior massa total acumulada no interior do ciclone).

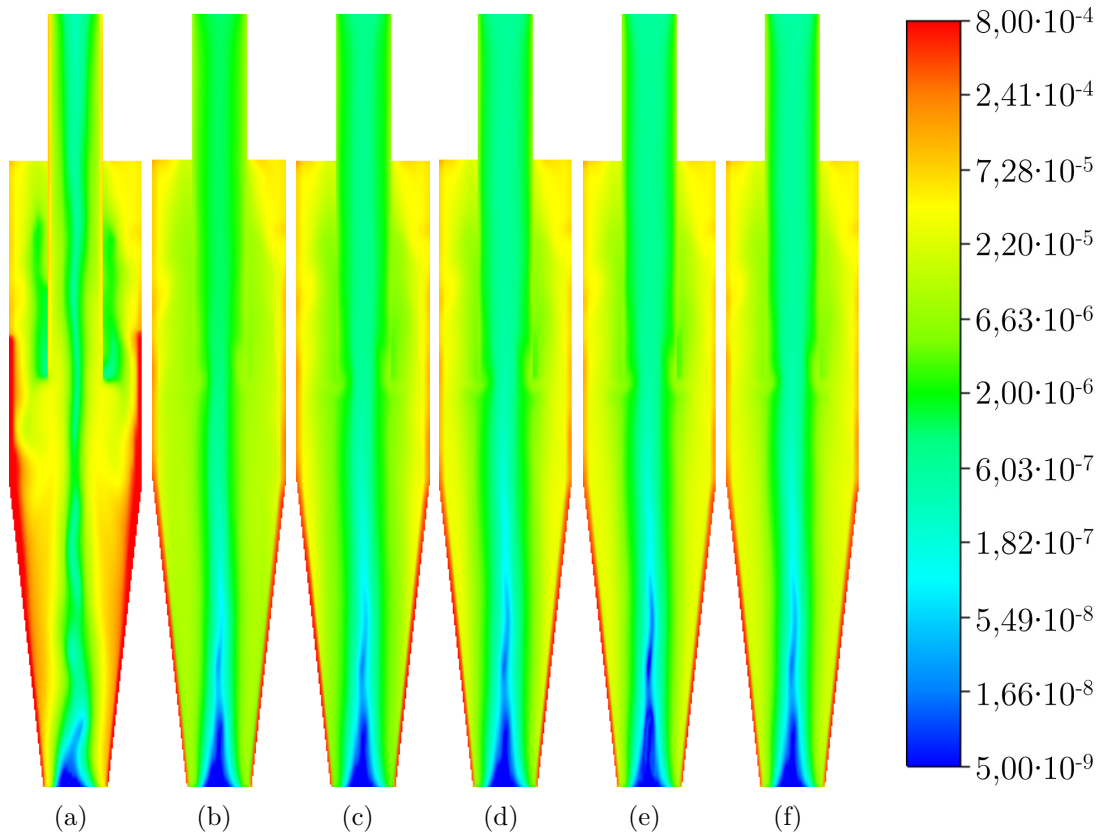


Figura 32 – Mapas da fração volumétrica média da fase sólida no ciclone para as simulações monodispersa e polidispersa com diferentes números de pontos de quadratura. (a) diâmetro médio; (b) dois pontos; (c) três pontos; (d) quatro pontos; (e) cinco pontos e (f) seis pontos de quadratura

Esta maior concentração de sólidos no ciclone pode ter diminuído o pico de velocidade tangencial da fase gasosa o que explicaria a menor perda de carga e eficiência

³ O erro relativo foi definido como $Erro = \frac{|\eta_{exp} - \eta_6|}{\eta_{exp}}$, na qual η_{exp} é a eficiência global experimental e η_6 é a eficiência global da simulação polidispersa com seis pontos de quadratura

global. Ademais, foi avaliado o efeito do número de pontos de quadratura sobre o perfil da velocidade tangencial da fase gasosa. A Figura 33 apresenta os perfis da velocidade tangencial na altura de 216 mm das simulações com dois, três, quatro, cinco e seis pontos de quadratura. Pela análise da Figura 33, percebe-se claramente que os perfis de velocidade apresentam o comportamento do vórtice de Rankine e que os diferentes pontos de quadratura apresentam perfis muito similares e convergentes, possuindo uma pequena diferença no pico da velocidade tangencial.

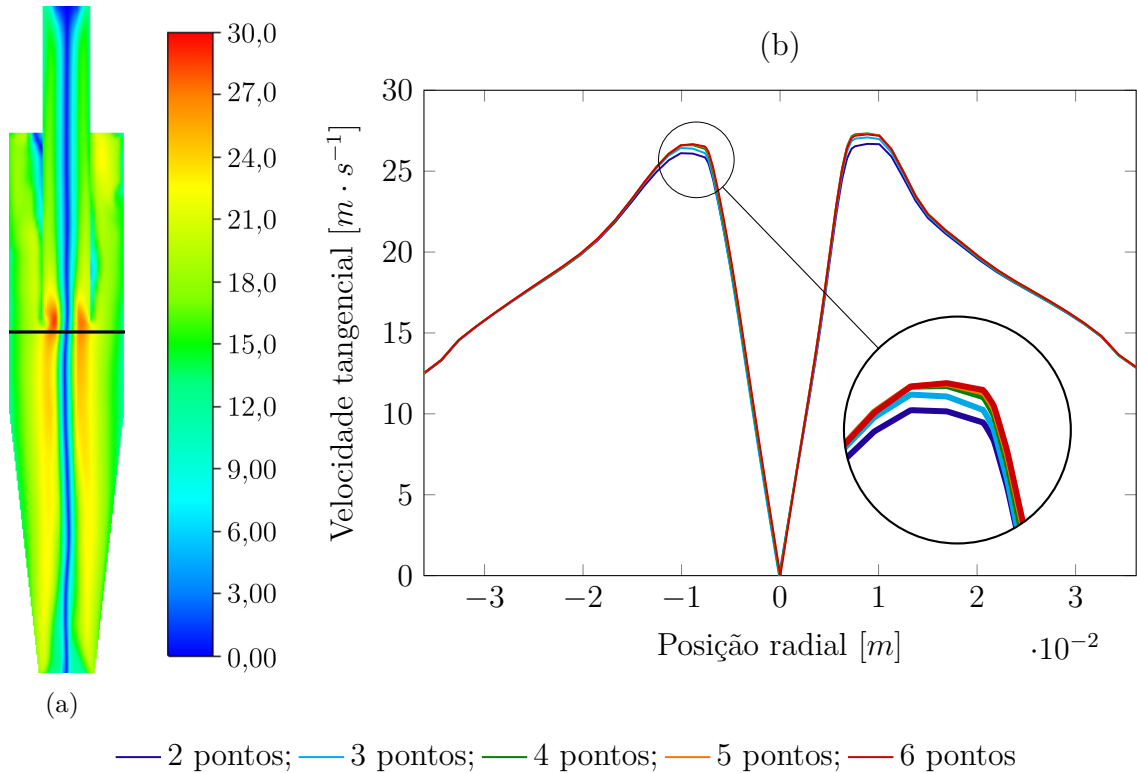


Figura 33 – Velocidade tangencial média da fase gasosa no ciclone com alta carga de sólidos. (a) mapa da velocidade tangencial média da fase gasosa para a simulação polidispersa com seis pontos de quadratura ressaltando a altura de 216 mm (b) velocidade tangencial média da fase gasosa em função da posição radial para as simulações polidispersas com diferentes pontos de quadratura na altura de 216 mm

Quando se compara, através da Figura 34, o perfil velocidade tangencial média da fase gasosa na altura de 216 mm com os resultados obtidos pelas simulações de Huang *et al.* (2018) e pelas simulações utilizando um modelo monodisperso e polidisperso com seis pontos de quadratura, nota-se que os resultados de Huang *et al.* (2018) apresentaram um pico de velocidade tangencial menor que o da simulação polidispersa. Já a simulação utilizando um modelo monodisperso apresentou um perfil de velocidade com valores menores que o do modelo polidisperso, sendo possível observar o achatamento do pico da velocidade tangencial, sendo aproximadamente a metade do valor do pico da velocidade do modelo polidisperso. Isto já era esperado tendo em vista o alta fração volumétrica de

sólidos na região central do ciclone, o que de certa forma explica a baixa perda de carga e eficiência de coleta.

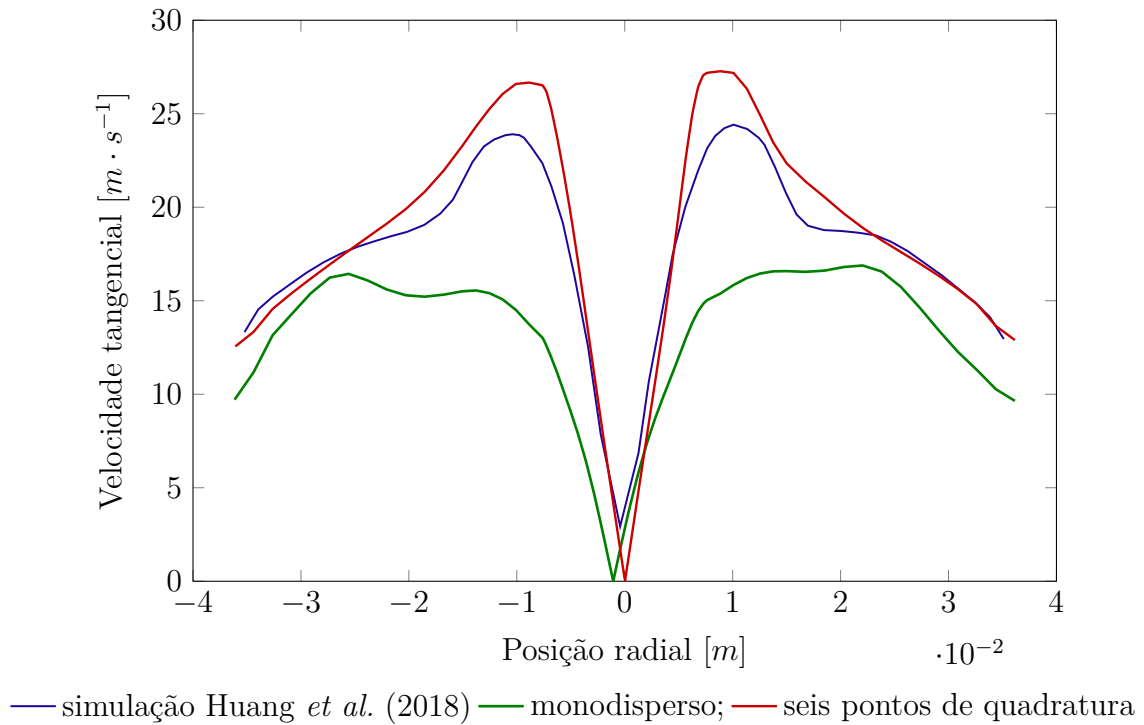


Figura 34 – Velocidade tangencial média pela posição radial do ciclone na altura de 216 mm comparando diferentes simulações

A partir da fração volumétrica dos pontos de quadraturas foram calculados os pesos e então os momentos da PDF. O diâmetro médio de Sauter (SMD) foi calculado dividindo o momento de ordem três pelo de ordem dois. Os mapas do SMD médio para as simulações polidispersas com dois, três, quatro, cinco e seis pontos de quadratura é apresentado na Figura 35.

Nota-se, pela Figura 35, há um aumento do diâmetro médio no sentido da parede do ciclone como era de se esperar uma vez que partículas com diâmetros maiores, através da atuação da força centrífuga gerada pelo escoamento em espiral, tendem a ir em direção da parede mais facilmente. De forma a avaliar quantitativamente o comportamento do diâmetro médio de Sauter, a Figura 36 apresenta os perfis do SMD na altura de 216 mm para as simulações polidispersas com dois, três, quatro, cinco e seis pontos de quadraturas.

Pela análise da Figura 36, observa-se, novamente, que o SMD aumenta no sentido da parede do ciclone. Salienta-se, ainda, que para a simulação polidispersa com dois pontos de quadratura que próximo a parede o SMD tende ao valor da maior abscissa, enquanto que no interior do ciclone observa-se que o SMD assumiu o valor da menor abscissa, de modo que, com o aumento do número de quadraturas observa-se convergência das curvas de diâmetro médio de Sauter na região de maior fração volumétrica de sólidos.

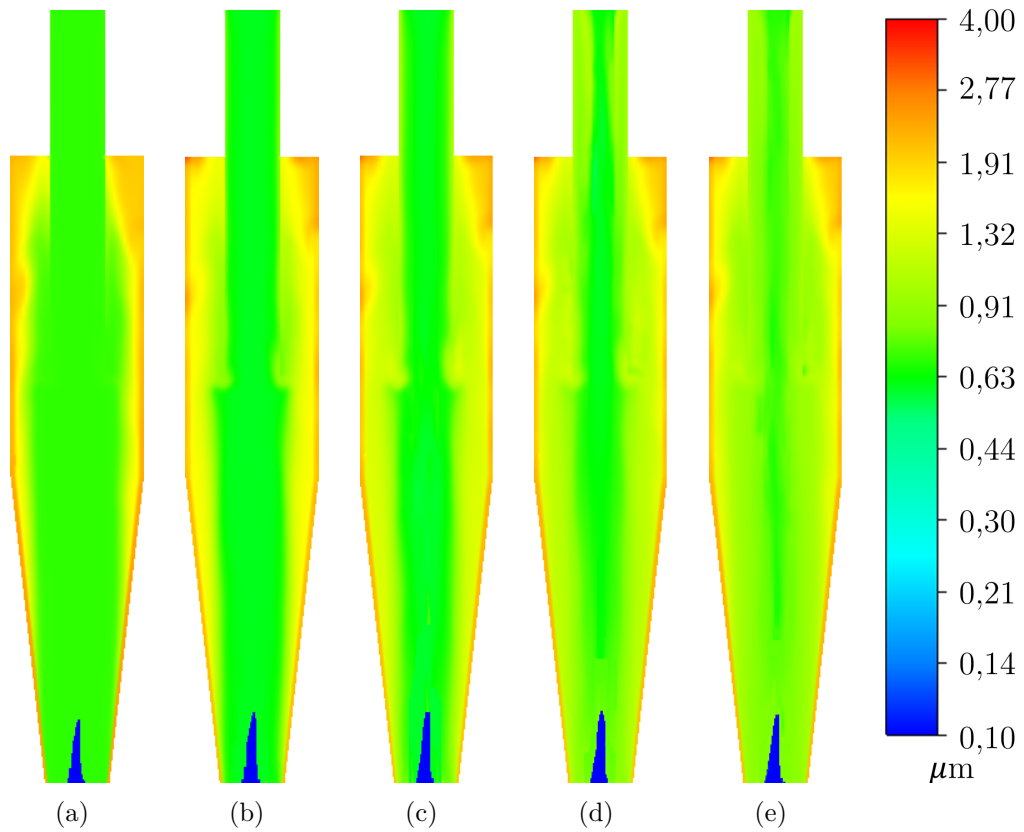


Figura 35 – Comparação do perfil médio do diâmetro médio de Sauter no ciclone para diferentes pontos de quadratura. (a) dois pontos; (b) três pontos; (c) quatro pontos; (d) cinco pontos e (e) seis pontos de quadratura

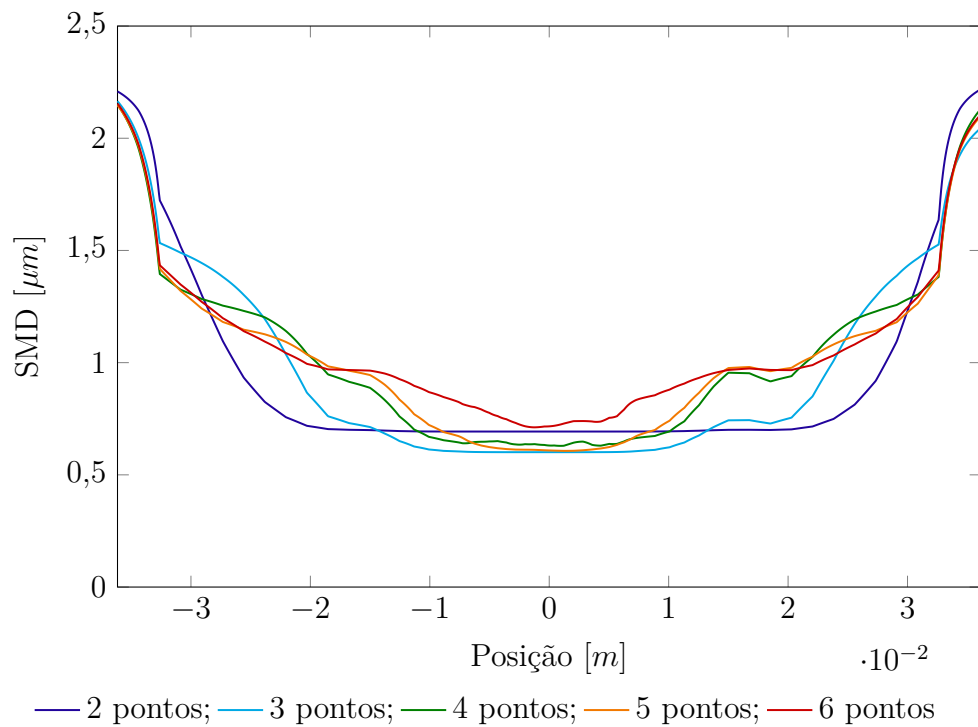


Figura 36 – Perfis do diâmetro médio de Sauter na altura de 216 mm das simulações polidispersas com diferentes números de pontos de quadratura

Com intuito de obter a curva de eficiência de coleta do ciclone, utilizou-se médias mássicas dos pesos na seção transversal da saída do ciclone para o *hopper* para calcular os momentos da função densidade de probabilidade de número. Utilizando estes momentos foram reconstruídas as distribuições do tamanho de partículas em massa para as simulações com dois, três, quatro, cinco e seis pontos de quadratura e estão apresentadas na Figura 37. Percebe-se que há uma convergência das curvas reconstruídas com o aumento do número de quadraturas. Além disso, pode-se observar pela Figura 17 que o diâmetro mínimo na entrada foi de $0.3 \mu m$, entretanto, as reconstruções apresentadas na Figura 37 apresentam probabilidades para diâmetros menores que $0.3 \mu m$. Conclui-se, portanto, que o método de reconstrução apresenta dificuldade em representar, especialmente, as caldas da distribuição uma vez que ela está relacionada com momentos de alta ordem.

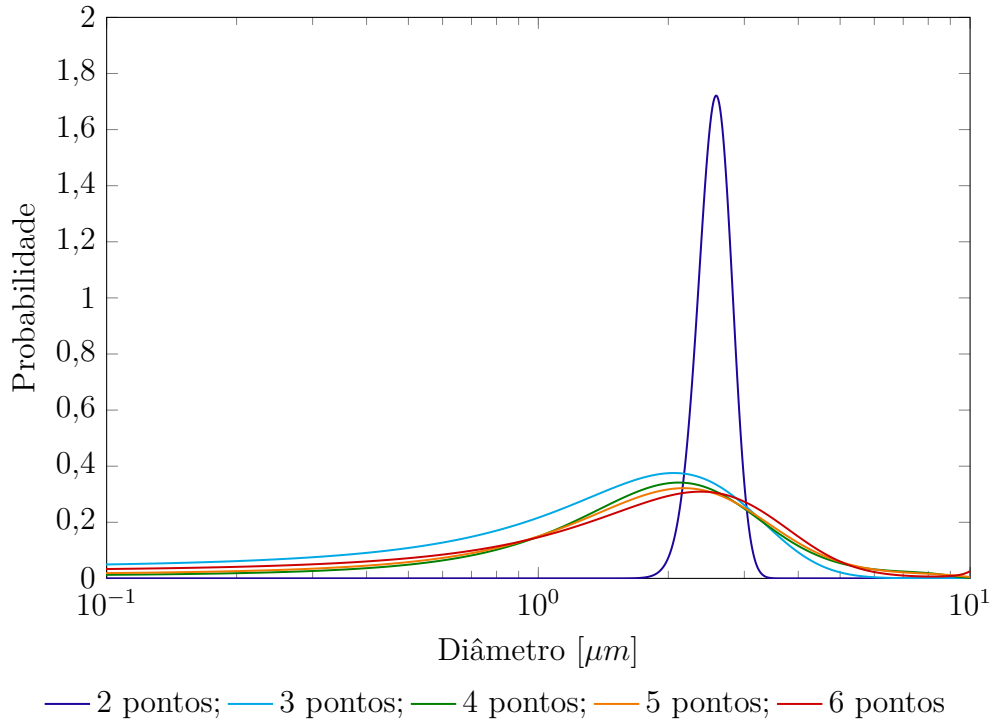


Figura 37 – Reconstrução da distribuição mássica do tamanho de partículas na saída do ciclone para o *hopper* dos modelos polidispersos com diferentes números de pontos de quadratura

Sendo assim, com base nessas reconstruções, na distribuição de partícula inicial e na eficiência global, foram calculadas as curvas de eficiência de coleta para as simulações polidispersas com dois, três, quatro, cinco e seis pontos de quadratura utilizando a Equação 2.7. A Figura 38 apresenta as curvas de eficiência. Foram calculados também os erros relativos⁴ das curvas de eficiência comparados à curva experimental e estão sintetizados na Tabela 19.

⁴ O erro relativo foi definido como $Erro = \frac{|\eta_{exp}(d) - \eta_q(d)|}{\eta_{exp}(d)}$, na qual $\eta_{exp}(d)$ é a curva de eficiência experimental e $\eta_q(d)$ é curva de eficiência para modelos polidispersos

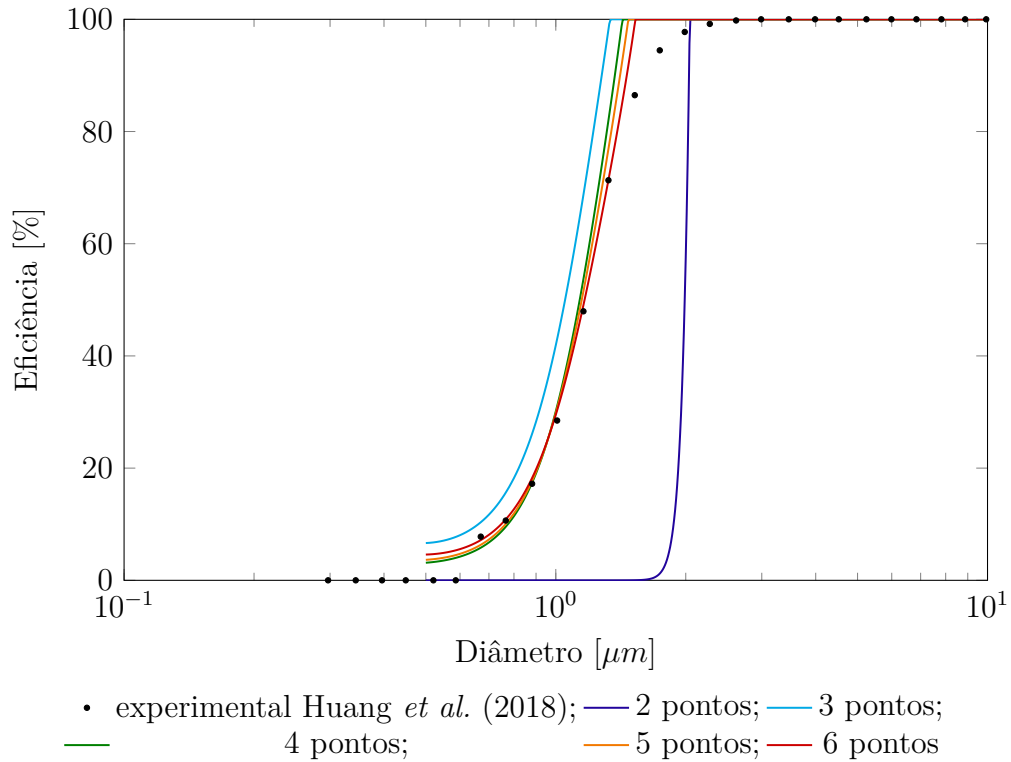


Figura 38 – Curva de eficiência do ciclone de alta carga considerando as partículas coletadas utilizando o método da máxima entropia para reconstrução da PDF para os diferentes números de pontos de quadratura

Tabela 19 – Erro relativo das curvas de eficiência para os diferentes pontos de quadraturas comparado com o resultado experimental de Huang *et al.* (2018)

Pontos de quadraturas	Erro relativo [%]
2 pontos	42,7
3 pontos	14,1
4 pontos	4,88
5 pontos	3,78
6 pontos	2,09

Pela Tabela 19 é possível observar que as curvas para 4, 5 e 6 pontos de quadratura ficaram bem próximas ao experimental possuindo um erro menor que 5%. Contudo, ao observar a Figura 38, percebe-se que a forma em “S” tradicional da curva de eficiência não foi obtida pelas reconstruções, de modo que não foi possível representar corretamente a região próxima à eficiência de 100%. Isto pode estar relacionado ao fato de que para reconstruir as caudas da distribuição são necessários momentos de ordens superiores, assim, com o aumento do número de pontos de quadratura pode ser possível representar o formato “S” da curva de eficiência.

No entanto, deve-se considerar o tempo de simulação adicional que o uso de um número maior de pontos de quadratura acarretará. Por exemplo, a simulação poli-

dispersa com seis pontos de quadratura apresentou aproximadamente o dobro do tempo computacional quando comparado com a simulação com quatro pontos de quadratura, tendo o primeiro consumido 74,3 h de processamento e o segundo 45,0 h. Deste modo, observa-se que o aumento do tempo computacional não é proporcional ao aumento do número de quadraturas e, assim, a escolha do número de pontos de quadraturas a ser utilizado depende do erro desejado em relação ao tempo computacional para obter os resultados da simulação.

Por fim, foi proposto na Figura 39 comparar a curva de eficiência obtida pela reconstrução utilizando seis pontos de quadratura, a eficiência dos pontos de quadratura, os resultados obtidos por Huang *et al.* (2018), que utilizaram uma abordagem Euleriana-Lagrangiana com um acoplamento de duas vias. Pode-se observar que apesar da curva de eficiência obtida pela simulação polidispersa com seis pontos de quadratura não possuir o formato “S”, a eficiência dos pontos de quadratura em si seguem esta tendência. Já a metodologia utilizada por Huang *et al.* (2018) apresentou uma eficiência maior que a experimental para a maioria dos diâmetros, representando melhor apenas a região próxima à eficiência de 100%.

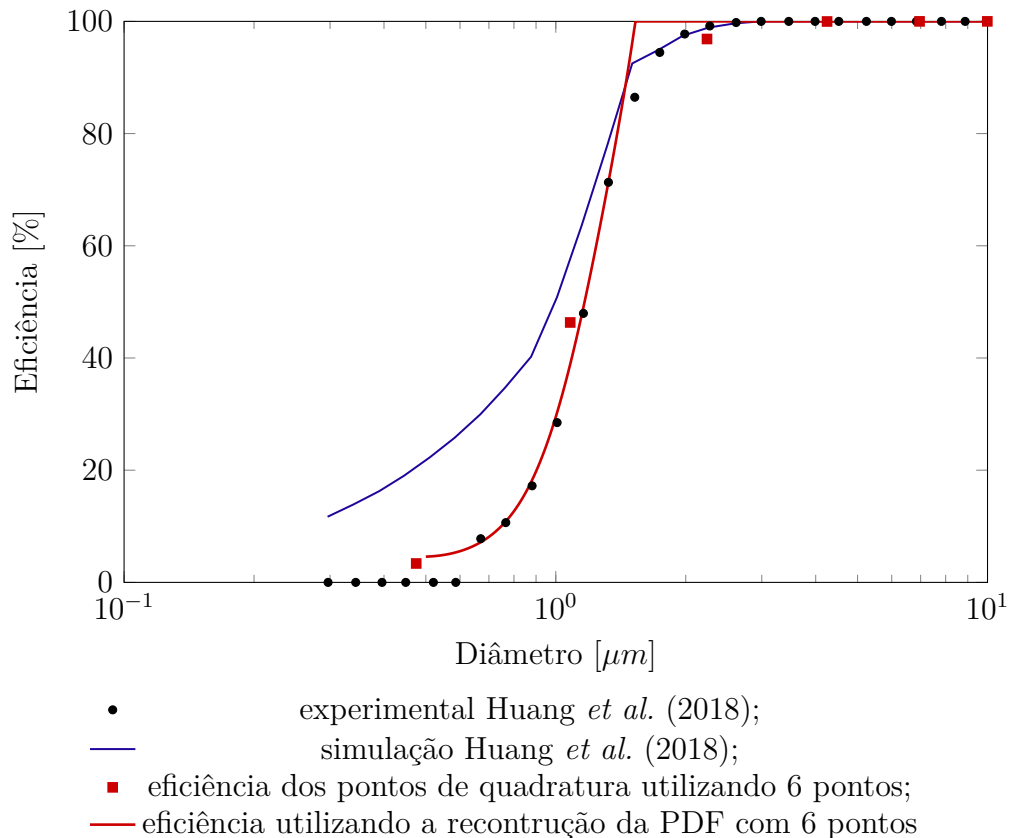


Figura 39 – Comparação da curva de eficiência do ciclone de alta carga com a simulação e dados experimentais de Huang *et al.* (2018)

4.3.3 Análise da Influência do Arraste Sólido-Sólido

De modo a avaliar a influência do arraste sólido-sólido, foram comparados os modelos de duas e quatro vias. A simulação utilizando um arraste de apenas duas vias apresentou uma queda de pressão de 709,33 Pa e uma eficiência global de coleta de 81,4%, uma diferença relativa de 3,6% em relação ao modelo polidisperso. Para calcular a curva de eficiência, primeiramente foi reconstruída a distribuição do tamanho de partículas em massa. Assim sendo, a Figura 40 apresenta as curvas reconstruídas para os modelos de duas e quatro vias.

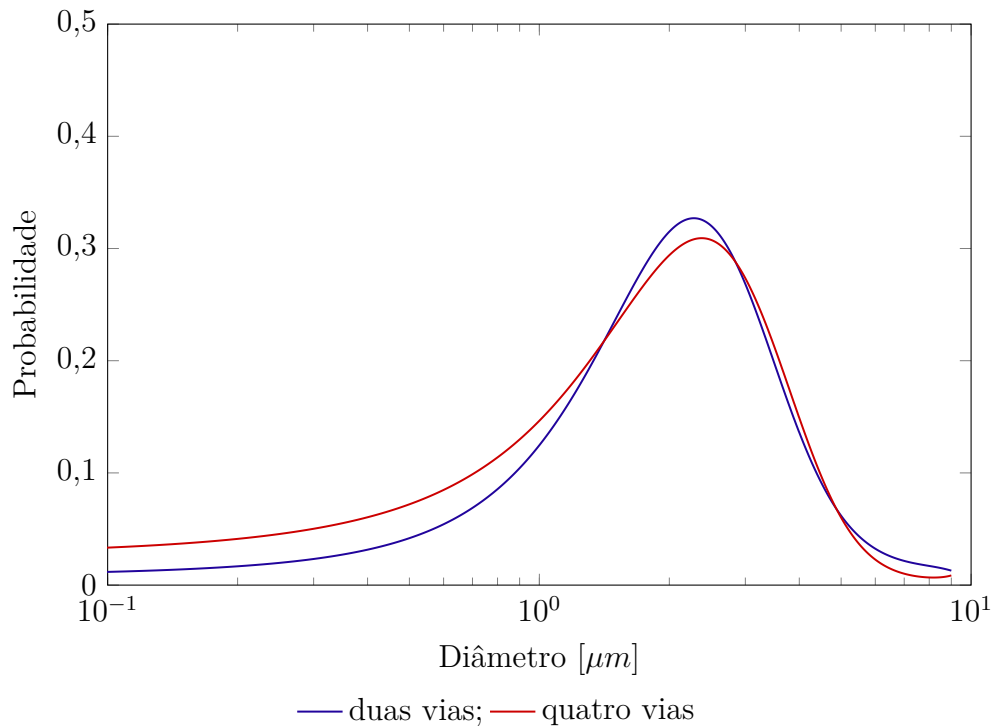


Figura 40 – Reconstrução da distribuição do tamanho de partículas em massa para o arraste de duas e quatro vias

Pode-se constatar pela Figura 40 que um arraste de duas vias apresentou uma menor probabilidade para diâmetro menores que 1 μm , demonstrando que o arraste sólido-sólido carregou as partículas de diâmetros menores para serem coletadas e assim aumentando a eficiência global. Deste modo, com base na reconstrução, a Figura 41 apresenta a curva de eficiência para as simulações utilizando o arraste de duas e quatro vias e também apresenta o resultado experimental obtido por Huang *et al.* (2018).

Pela Figura 41 pode-se observar que para partículas menores que 1,5 μm , a simulação utilizando um arraste de duas vias apresentou uma menor eficiência de coleta, sendo o erro relativo desta curva comparado com a experimental de 8,57%, o qual é maior que o erro da simulação com arraste de quatro vias. Apesar de parecer que as curvas são muito próximas, analisando por exemplo a eficiência do diâmetro de 1 μm , para

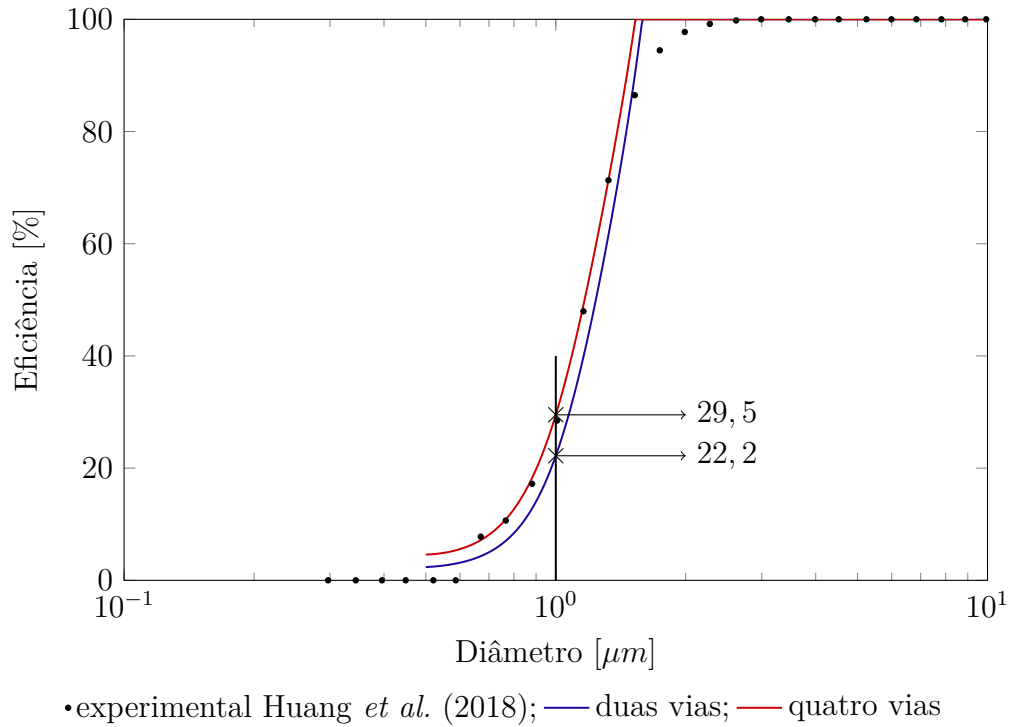


Figura 41 – Curva de eficiência do ciclone de alta carga considerando as partículas coletadas utilizando o método da máxima entropia para reconstrução da PDF para os modelos de arraste de duas e quatro vias

a simulação utilizando o arraste de duas vias o valor foi de 22,2% enquanto que o de quatro vias foi de 29,5%, possuindo um erro relativo de aproximadamente 25%. Conclui-se que o arraste sólido-sólido tem efeito significativo no comportamento das partículas, especialmente, as de menores diâmetros.

4.3.4 Avaliação de Outras Condições Operacionais

No intuito de avaliar a metodologia para outros casos, variou-se a velocidade de entrada e a concentração de sólidos. A Tabela 20 apresenta os resultados da queda de pressão e eficiência global de coleta das simulações dos casos apresentados na seção 3.2.6.4, os resultados experimentais da eficiência global de coleta obtidos por Huang *et al.* (2018) e a eficiência global de coleta calculada utilizando o modelo proposto por Massarani (1997).

Pode-se constatar pela Tabela 20, que o aumento da velocidade resultou no aumento da queda de pressão seguindo a proporcionalidade do número de Euler. Além disso, pode-se salientar ainda que o aumento da carga de sólidos acarretou no aumento da queda de pressão. Uma explicação, é que a queda de pressão gerada pelo aumento da concentração de partículas foi maior do que o efeito desta na atenuação da velocidade tangencial do escoamento. Esta atenuação pode ser observada na Figura 42 que apresenta a velocidade tangencial da fase gasosa na altura de 216 mm para os casos 1 e 2.

Tabela 20 – Resultados da queda de pressão e eficiência global de coleta das simulações dos casos apresentados na seção 3.2.6.4 e os resultados experimentais da eficiência global de coleta obtidos por Huang *et al.* (2018)

Casos	Queda de pressão [Pa]	Eficiência global [%]	Eficiência global experimental de Huang <i>et al.</i> (2018) [%]	Eficiência global calculada por Massarani (1997) [%]
1	711,55	84,4	84,2	71,87
2	677,13	85,2	83,7	71,88
3	1018,5	90,9	85,4	76,21
4	989,82	91,2	85,8	76,22

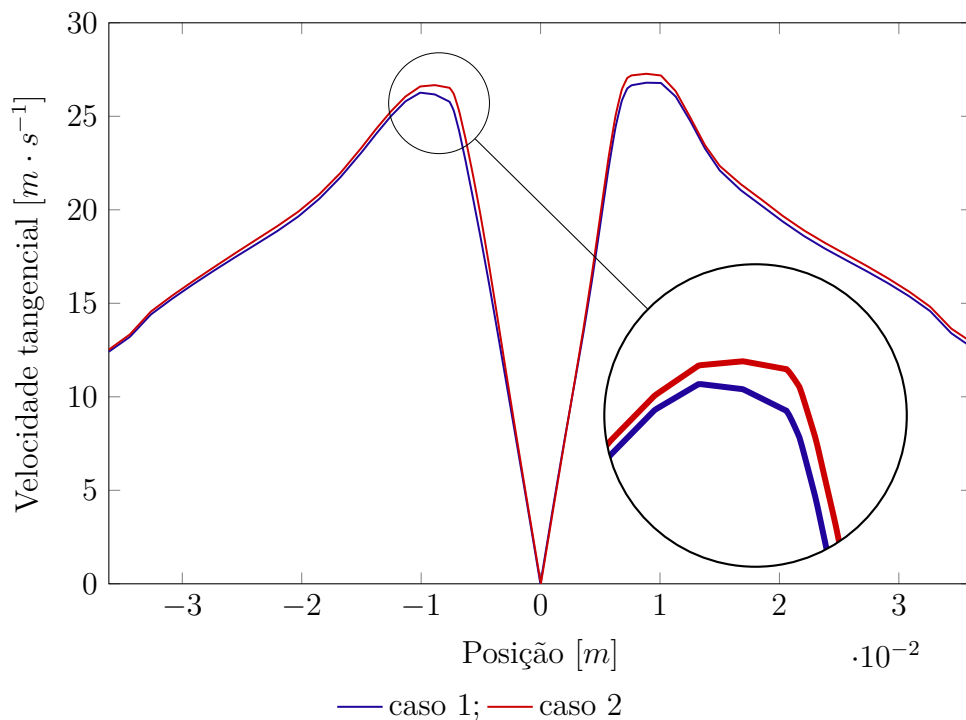


Figura 42 – Perfis da velocidade tangencial média da fase gasosa na altura de 216 mm para os casos 1 e 2

Percebe-se ainda pela Tabela 20, que a eficiência global dos casos 3 e 4, que possuem uma velocidade de entrada de $18 \text{ m} \cdot \text{s}^{-1}$, foram maiores em comparação com o experimental de Huang *et al.* (2018). Observa-se que para os resultados experimentais que o aumento da velocidade resultou em um menor aumento da eficiência global quando comparado ao resultado obtido por este trabalho e a eficiência calculado utilizando o modelo de Massarani (1997), o que pode sugerir problemas experimentais. Além disso, pode-se perceber que tanto para a simulação polidispersa quanto utilizando a correlação de Massarani (1997) para o cálculo da eficiência o fato de que o aumento da carga de sólidos diminuiu a eficiência global de coleta, o que vai ao encontro dos resultados obtidos pela simulação de Huang *et al.* (2018) mas o que não foi observado por eles experimentalmente, principalmente, para a velocidade de entrada de $15 \text{ m} \cdot \text{s}^{-1}$.

A partir da eficiência global e reconstrução das curvas de distribuição do diâmetro das partículas coletadas foram calculadas as curvas de eficiência. A Figura 43 apresenta as curvas de eficiência para os casos 1 (a), 2 (b), 3 (c) e 4 (d) e também as curvas experimentais de Huang *et al.* (2018). Pode-se observar que as curvas de eficiência das simulações polidispersas com seis pontos de quadratura foram próximas às curvas experimentais. Pode-se verificar que as Figuras 43 (c) e (d), as quais são os resultados da simulação com velocidade de entrada de $18 \text{ m} \cdot \text{s}^{-1}$, que as curvas estão deslocadas para a esquerda em relação aos dados experimentais. Isto é decorrente do fato de a eficiência global das simulações terem sido maiores que as experimentais.

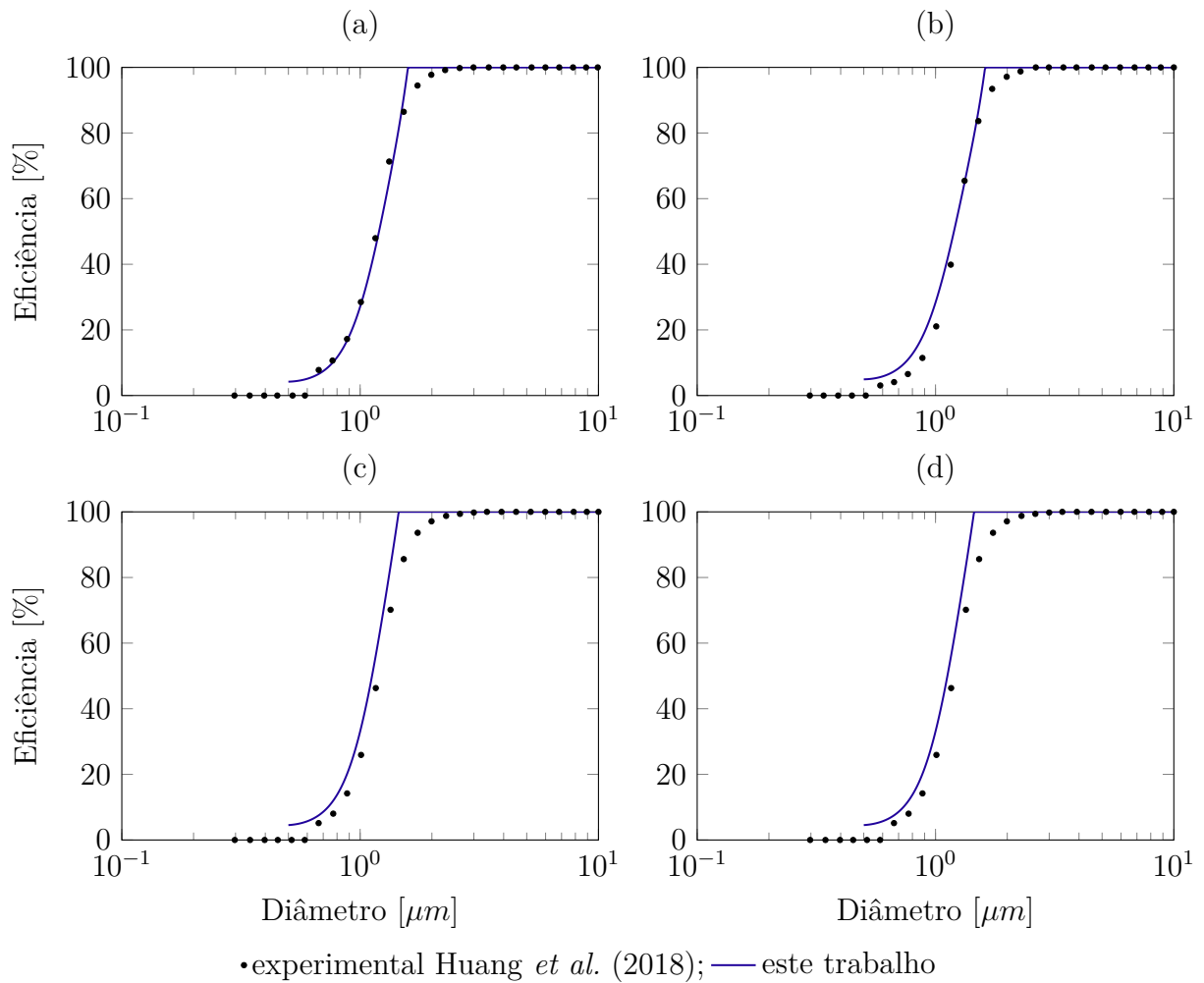


Figura 43 – Curva de eficiência de coleta para os casos: (a) 1; (b) 2; (c) 3 e (d) 4

Desta forma, empregando a metodologia proposta, foi possível construir a curva de eficiência destes diferentes casos variando a velocidade de entrada e a carga de sólidos.

5 CONCLUSÃO

Este trabalho propôs desenvolver um modelo Euleriano-Euleriano acoplado às equações de balanço populacional resolvidas pelo método DQMOM com objetivo de reconstruir a curva de eficiência de coleta de partículas em um ciclone operando a alta carga de sólidos utilizando o *software* comercial Fluent 17.2. Desta forma, através de todas as análises feitas durante este estudo foram possíveis obter as seguintes conclusões:

- o código foi verificado comparando-o com o trabalho de Niemi (2012), sendo obtido resultados próximos;
- o método de máxima entropia necessita de um menor número de parâmetros de entrada e apresentou menores erros quando comparados ao método da *spline* cúbica, de modo que foram obtidos erros menores que 3% quando utilizados pelo menos 7 momentos e, assim, para reconstruções mais precisas deve utilizar pelo menos quatro pontos de quadratura;
- utilizando um modelo polidisperso com quatro pontos de quadratura foi possível representar o diâmetro médio de Sauter e a dispersão da PDF com um erro aceitável. No entanto, quando se deseja avaliar a obliquidade e a curtose da PDF necessita-se de um maior número de quadraturas, sendo observado resultados mais condizentes para seis pontos de quadratura.
- os mapas da fração volumétrica média e os perfis de velocidade tangencial média demonstraram que os resultados obtidos para as simulações de ciclone operando a alta carga de sólidos utilizando diferentes números de quadraturas foram muito próximos. Contudo, esses resultados foram muito diferentes do obtido pela simulação monodispersa. De modo que, como as eficiências globais das simulações polidispersas foram muito mais próxima da experimental, pode-se dizer que a metodologia utilizada obteve resultados mais condizentes do que a modelagem tradicional;
- as curvas de eficiência calculadas a partir da reconstrução da distribuição mássica do tamanho de partículas para 4, 5 e 6 pontos de quadraturas tiveram um erro menor que 5% comparada com a curva experimental. Resultado este muito melhor que o obtido por Huang *et al.* (2018) utilizando uma abordagem Euleriana-Lagrangiana com acoplamento de duas vias;
- o arraste de quatro vias proporcionou uma eficiência de coleta mais próxima da experimental, sendo que o de duas subestima a eficiência;

- a modelagem utilizada neste estudo se mostrou promissora pra calcular a curva de eficiência de coleta para as diferentes condições de velocidade de entrada e carga de sólidos.

Além das conclusões obtidas das análises realizadas, também foi possível levantar alguns tópicos como sugestão para trabalhos futuros listados abaixo:

- realizar um estudo experimental para obter curvas de distribuição de tamanho de partículas coletadas no intuito de comparar com as curvas reconstruídas após a simulação do ciclone;
- considerar a quebra de partículas nas equações de balanço populacional e, assim, acrescentar novas equações de transporte para as abscisas, de modo que elas passem a variar no ciclone, sendo possível, assim, calcular a curva de eficiência na saída do *vortex finder*;
- avaliar modelos de arraste tanto para as interações gás-sólido quanto para a sólido-sólido;
- avaliar se número de quadraturas maiores do que seis geram diferenças significativas nos resultados já obtidos.

REFERÊNCIAS

ANSYS, I. *Release 17.2, Help System, Fluent Customization Manual*, ANSYS®, Inc. 2016.

ASHRY, Y. E.; ABDELRAZEK, A. M.; ELSHORBAGY, K. A. Numerical and experimental study on the effect of solid particle sphericity on cyclone pressure drop. *Separation Science and Technology*, Taylor & Francis, v. 53, n. 15, p. 2500–2516, 2018.

BALESTRIN, E.; DECKER, R. K.; NORILER, D.; BASTOS, J. C. S. C.; MEIER, H. F. An alternative for the collection of small particles in cyclones: Experimental analysis and CFD modeling. *Separation and Purification Technology*, Elsevier, v. 184, p. 54–65, 2017.

BARTH, W. Design and layout of the cyclone separator on the basis of new investigations. *Brenn. Warne Kraft*, v. 8, n. 1, p. 1–9, 1956.

BERNARDO, S.; MORI, M.; PERES, A. P.; DIONISIO, R. P. 3-d computational fluid dynamics for gas and gas-particle flows in a cyclone with different inlet section angles. *Powder Technology*, Elsevier, v. 162, n. 3, p. 190–200, 2006.

BOYSAN, F.; AYERS, W. H.; SWITHEBANK, J. A fundamental mathematical modelling approach to cyclone design. *Transactions of the institution of chemical engineers*, v. 60, n. 4, p. 222–230, 1982.

BRAR, L. S.; SHARMA, R.; ELSAYED, K. The effect of the cyclone length on the performance of stairmand high-efficiency cyclone. *Powder Technology*, Elsevier, v. 286, p. 668–677, 2015.

CASAL, J.; MARTINEZ-BENET, J. M. A better way to calculate cyclone pressure drop. *Chemical Engineering*, p. 99–100, 1983.

CELIK, I. B.; GHIA, U.; ROACHE, P. J. Procedure for estimation and reporting of uncertainty due to discretization in cfd applications. *Journal of fluids Engineering-Transactions of the ASME*, v. 130, n. 7, 2008.

CHIEN, S.-F. *et al.* Settling velocity of irregularly shaped particles. *SPE Drilling & Completion*, Society of Petroleum Engineers, v. 9, n. 04, p. 281–289, 1994.

CHOI, M.; LEE, S.; HAM, J.-K. Study on the configuration effect of a vortex finder for a gas-solid cyclone separator using eulerian-eulerian approach. In: AMERICAN SOCIETY OF MECHANICAL ENGINEERS. *ASME 2013 International Mechanical Engineering Congress and Exposition*. San Diego, 2013. v. 7A, p. 1–10.

COSTA, K. K.; JR, O. L. S.; DECKER, R. K.; REINEHR, E. L.; MARTIGNONI, W. P.; MEIER, H. F. Effects of phases' numbers and solid-solid interactions on the numerical simulations of cyclones. *Chemical Engineering Transactions*, v. 32, p. 1567–1572, 2013.

DEMIR, S.; KARADENIZ, A.; AKSEL, M. Effects of cylindrical and conical heights on pressure and velocity fields in cyclones. *Powder Technology*, Elsevier, v. 295, p. 209–217, 2016.

- DERKSEN, J. J. Separation performance predictions of a stairmand high-efficiency cyclone. *AIChE Journal*, Wiley Online Library, v. 49, n. 6, p. 1359–1371, 2003.
- DERKSEN, J. J.; AKKER, H. E. A. Van den; SUNDARESAN, S. Two-way coupled large-eddy simulations of the gas-solid flow in cyclone separators. *AIChE Journal*, Wiley Online Library, v. 54, n. 4, p. 872–885, 2008.
- DUTTA, A.; CONSTALES, D.; HEYNDERICKX, G. J. Applying the direct quadrature method of moments to improve multiphase fcc riser reactor simulation. *Chemical engineering science*, Elsevier, v. 83, p. 93–109, 2012.
- EL-BATSH, H. M. Improving cyclone performance by proper selection of the exit pipe. *Applied mathematical modelling*, Elsevier, v. 37, n. 7, p. 5286–5303, 2013.
- ERGUN, S. Fluid flow through packed columns. *Chem. Eng. Prog.*, v. 48, p. 89–94, 1952.
- FAN, R.; MARCHISIO, D. L.; FOX, R. O. Application of the direct quadrature method of moments to polydisperse gas–solid fluidized beds. *Powder technology*, Elsevier, v. 139, n. 1, p. 7–20, 2004.
- FATAHIAN, H.; FATAHIAN, E.; NIMVARI, M. E. Improving efficiency of conventional and square cyclones using different configurations of the laminarizer. *Powder technology*, Elsevier, v. 339, p. 232–243, 2018.
- FIRST, M. W. Cyclone dust collector design. In: *Am. Soc. Mech. Eng.* [S.l.: s.n.], 1949. v. 49, p. 127–132.
- GIDASPOW, D. *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. San Diego: Academic press, 1994.
- GORDON, R. G. Error bounds in equilibrium statistical mechanics. *Journal of Mathematical Physics*, AIP, v. 9, n. 5, p. 655–663, 1968.
- HAIDER, A.; LEVENSPIEL, O. Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder technology*, Elsevier, v. 58, n. 1, p. 63–70, 1989.
- HOFFMANN, A. C.; STEIN, L. E. *Gas Cyclones and Swirl Tubes: Principles, Design and Operation*. Berlin: Springer, 2002.
- HUANG, A.-N.; ITO, K.; FUKASAWA, T.; FUKUI, K.; KUO, H.-P. Effects of particle mass loading on the hydrodynamics and separation efficiency of a cyclone separator. *Journal of the Taiwan Institute of Chemical Engineers*, Elsevier, 2018.
- HUANG, A.-N.; MAEDA, N.; SHIBATA, D.; FUKASAWA, T.; YOSHIDA, H.; KUO, H.-P.; FUKUI, K. Influence of a laminarizer at the inlet on the classification performance of a cyclone separator. *Separation and Purification Technology*, Elsevier, v. 174, p. 408–416, 2017.
- JAMES, B. R. *Probabilidade: um curso em nível intermediário*. 3. ed. Rio de Janeiro: Instituto de Matematica Pura e Aplicada, 2010. 299 p.
- JOHN, V.; ANGELOV, I.; ÖNCÜL, A. A.; THÉVENIN, D. Techniques for the reconstruction of a distribution from a finite number of its moments. *Chemical Engineering Science*, Elsevier, v. 62, n. 11, p. 2890–2904, 2007.

- KARIMI, M.; AKDOGAN, G.; DELLIMORE, K. H.; BRADSHAW, S. M. Quantification of numerical uncertainty in computational fluid dynamics modelling of hydrocyclones. *Computers & Chemical Engineering*, Elsevier, v. 43, p. 45–54, 2012.
- KASPER, G. Dynamics and measurement of smokes. i size characterization of nonspherical particles. *Aerosol Science and Technology*, Taylor & Francis, v. 1, n. 2, p. 187–199, 1982.
- KOZOŁUB, P.; KLIMANEK, A.; BIAŁECKI, R. A.; ADAMCZYK, W. P. Numerical simulation of a dense solid particle flow inside a cyclone separator using the hybrid Euler–Lagrange approach. *Particuology*, Elsevier, v. 31, p. 170–180, 2017.
- LAPPLE, C. E. Gravity and centrifugal separation. *American Industrial Hygiene Association Quarterly*, Taylor & Francis Group, v. 11, n. 1, p. 40–48, 1950.
- LAUNDER, B. E.; SPALDING, D. B. The numerical computation of turbulent flows. *Computer methods in applied mechanics and engineering*, Elsevier, v. 3, n. 2, p. 269–289, 1974.
- LEBAZ, N.; COCKX, A.; SPÉRANDIO, M.; MORCHAIN, J. Reconstruction of a distribution from a finite number of its moments: a comparative study in the case of depolymerization process. *Computers & Chemical Engineering*, Elsevier, v. 84, p. 326–337, 2016.
- LIEN, F.-S.; LESCHZINER, M. Assessment of turbulence-transport models including non-linear rng eddy-viscosity formulation and second-moment closure for flow over a backward-facing step. *Computers & Fluids*, Elsevier, v. 23, n. 8, p. 983–1004, 1994.
- LIN, F.; FLOOD, A. E.; MELESHKO, S. V. Exact solutions of population balance equation. *Communications in Nonlinear Science and Numerical Simulation*, Elsevier, v. 36, p. 378–390, 2016.
- LUN, C. K. K.; SAVAGE, S. B.; JEFFREY, D. J.; CHEPURNIY, N. Kinetic theories for granular flow: inelastic particles in couette flow and slightly inelastic particles in a general flowfield. *Journal of fluid mechanics*, Cambridge University Press, v. 140, p. 223–256, 1984.
- MANNINEN, M.; TAIVASSALO, V.; KALLIO, S. *On the mixture model for multiphase flow*. Technical Research Centre of Finland Finland: VTT Publications 288, 1996.
- MARIANI, F.; RISI, F.; GRIMALDI, C. N. Separation efficiency and heat exchange optimization in a cyclone. *Separation and Purification Technology*, Elsevier, v. 179, p. 393–402, 2017.
- MASSARANI, G. *Fluidodinâmica em sistemas particulados*. Rio de Janeiro: Editora UFRJ, 1997.
- MAZZEI, L. *Eulerian modelling and computational fluid dynamics simulation of mono and polydisperse fluidized suspensions*. 100 f. Tese (Doutorado em Engenharia Química), University College London, Department of Chemical Engineering, Londres, UK, 2008.

- MEIER, H. F. *Modelagem Fenomenológica e Simulação Bidimensional de Ciclones por Técnicas da Fluidodinâmica Computacional*. 105 f. Tese (Doutorado em Engenharia Química), Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, 1998.
- MOHAMMAD-DJAFARI, A. A matlab program to calculate the maximum entropy distributions. In: SMITH, C. R.; ERICKSON, G. J.; NEUDORFER, P. O. (Ed.). *Maximum Entropy and Bayesian Methods*. Seattle: Springer, 1991. p. 221–233.
- NIEMI, T. *Particle Size Distribution in CFD Simulation of Gas-Particle Flows*. 42 f. Dissertação (Mestrado em Tecnologia), School of Science, Aalto University, Finland, 2012.
- NORILER, D.; VEGINI, A.; SOARES, C.; BARROS, A.; MEIER, H.; MORI, M. A new role for reduction in pressure drop in cyclones using computational fluid dynamics techniques. *Brazilian Journal of Chemical Engineering*, SciELO Brasil, v. 21, n. 1, p. 93–101, 2004.
- OGAWA, A. Mechanical separation process and flow patterns of cyclone dust collectors. *Applied Mechanics Reviews*, American Society of Mechanical Engineers, v. 50, n. 3, p. 97–130, 1997.
- PELTOLA, J.; KALLIO, S.; HONKANEN, M.; SAARENINNE, P. Image based measurement of particle phase reynolds stresses in a laboratory scale circulating fluidized bed. In: *7th International Conference on Multiphase Flow - ICMF2010*. Tampa: USA, 2010.
- QIAN, F.; HUANG, Z.; CHEN, G.; ZHANG, M. Numerical study of the separation characteristics in a cyclone of different inlet particle concentrations. *Computers & chemical engineering*, Elsevier, v. 31, n. 9, p. 1111–1122, 2007.
- RAMKRISHNA, D. *Population Balances: Theory and Applications to Particulate Systems in Engineering*. San Diego: Academic Press, 2000.
- ROACHE, P. J. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, American Society of Mechanical Engineers, v. 116, n. 3, p. 405–413, 1994.
- RODI, W. Examples of turbulence models for incompressible flows. *AIAA Journal*, v. 20, n. 7, p. 872–879, 1982.
- ROTTA, J. C. Statistical theory of nonhomogeneous turbulence. *Z. Physik*, v. 129, p. 547–559, 1951.
- SCHAEFFER, D. G. Instability in the evolution equations describing incompressible granular flow. *Journal of differential equations*, Elsevier, v. 66, n. 1, p. 19–50, 1987.
- SCHMIDT, S.; BLACKBURN, H. M.; RUDMAN, M. Impact of outlet boundary conditions on the flow properties within a cyclone. In: *15th Australasian Fluid Mechanics Conference*. Sydney: Australia, 2004. p. 13–17.
- SCOPUS. *Scopus - Document search*. 2019. Acesso em: 07 jan. 2019. Disponível em: <<https://www.scopus.com/search/form.uri?display=basic&clear=t&origin=searchauthorlookup&txGid=0a6a1eb37e98e06a0ed3acfe80ab2012>>.

- SEDREZ, T. A.; DECKER, R. K.; SILVA, M. K. da; NORILER, D.; MEIER, H. F. Experiments and cfd-based erosion modeling for gas-solids flow in cyclones. *Powder Technology*, Elsevier, v. 311, p. 120–131, 2017.
- SGROTT, O. L.; NORILER, D.; WIGGERS, V. R.; MEIER, H. F. Cyclone optimization by complex method and cfd simulation. *Powder Technology*, Elsevier, v. 277, p. 11–21, 2015.
- SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, Nokia Bell Labs, v. 27, n. 4, p. 623–656, 1948.
- SHEPHERD, C. B.; LAPPLE, C. E. Flow pattern and pressure drop in cyclone dust collectors cyclone without intel vane. *Industrial & Engineering Chemistry*, ACS Publications, v. 32, n. 9, p. 1246–1248, 1940.
- SOUZA, F. J. de; SALVO, R. de V.; MARTINS, D. de M. Effects of the gas outlet duct length and shape on the performance of cyclone separators. *Separation and Purification Technology*, Elsevier, v. 142, p. 90–100, 2015.
- SOUZA, L. G. M. D.; JANIGA, G.; JOHN, V.; THÉVENIN, D. Reconstruction of a distribution from a finite number of moments with an adaptive spline-based algorithm. *Chemical Engineering Science*, Elsevier, v. 65, n. 9, p. 2741–2750, 2010.
- SPEZIALE, C. G.; SARKAR, S.; GATSKI, T. B. Modelling the pressure–strain correlation of turbulence: an invariant dynamical systems approach. *Journal of fluid mechanics*, Cambridge University Press, v. 227, p. 245–272, 1991.
- STAIRMAND, C. J. Pressure drop in cyclone separators. *Engineering*, v. 168, n. 4369, p. 409–412, 1949.
- SUN, X.; KIM, S.; YANG, S. D.; KIM, H. S.; YOON, J. Y. Multi-objective optimization of a stairmand cyclone separator using response surface methodology and computational fluid dynamics. *Powder Technology*, Elsevier, v. 320, p. 51–65, 2017.
- SWAMEE, P. K.; OJHA, C. S. P. Drag coefficient and fall velocity of nonspherical particles. *Journal of Hydraulic Engineering*, American Society of Civil Engineers, v. 117, n. 5, p. 660–667, 1991.
- SYAMLAL, M. *The particle-particle drag term in a multiparticle model of fluidization*. Technical report, National Technical Information Service, Springfield, VA, 1987.
- SYAMLAL, M.; ROGERS, W.; O'BRIEN, T. J. *MFIX Documentation: Volume 1, Theory Guide*. Springfield, VA: National Technical Information Service, 1993.
- WASILEWSKI, M.; BRAR, L. S. Optimization of the geometry of cyclone separators used in clinker burning process: A case study. *Powder Technology*, Elsevier, v. 313, p. 293–302, 2017.
- WEN, C.; YU, Y. Mechanics of fluidization. *Chem. Eng. Prog. Symp. Ser.*, v. 62, p. 100–111, 1966.
- YEOH, G. H.; CHEUNG, C. P.; TU, J. Chapter 3 - population balance approach — a generic framework. In: YEOH, G. H.; CHEUNG, C. P.; TU, J. (Ed.). *Multiphase Flow Analysis Using Population Balance Modeling*. Oxford: Butterworth-Heinemann, 2014. p. 69 – 90.

YUAN, C.; KONG, B.; PASSALACQUA, A.; FOX, R. O. An extended quadrature-based mass-velocity moment model for polydisperse bubbly flows. *The Canadian Journal of Chemical Engineering*, Wiley Online Library, v. 92, n. 12, p. 2053–2066, 2014.

ZHOU, F.; SUN, G.; ZHANG, Y.; CI, H.; WEI, Q. Experimental and cfd study on the effects of surface roughness on cyclone performance. *Separation and Purification Technology*, Elsevier, v. 193, p. 175–183, 2018.

Apêndices

APÊNDICE A – IMPLEMENTAÇÃO DO BALANÇO POPULACIONAL

O código define os termos transiente e convectivo da PBE resolvida pelo método DQMOM (Equação (3.29)) por meio das sub-rotinas *DEFINE_UDS_UNSTEADY* e *DEFINE_UDS_FLUX*. No entanto, a última sub-rotina necessita das velocidades dos pontos de quadratura, as quais foram calculadas a partir da resolução do sistema não linear gerado pela Equação (3.50). Esse sistema foi resolvido utilizando o método de Newton e implementado através da sub-rotina *DEFINE_ADJUST* e outras criadas para facilitar a implementação do método de Newton para sistemas não lineares. O diâmetro característico foi atualizado para cada célula computacional a cada iteração utilizando a sub-rotina *DEFINE_PROPERTY*. Por fim, a correção da fração volumétrica foi feita utilizando a sub-rotina *DEFINE_EXECUTE_AT_END*.

Ademais, é importante ressaltar que no cabeçalho do programa os seguintes pacotes foram incluídos “math.h”, “udf.h” e “linalg.h”. O primeiro é para habilitar algumas operações matemáticas, o segundo é para o Fluent compilar as novas sub-rotinas e o último possibilita utilizar a função *Gauss_Elimination_dp*. Posteriormente, foram definidas constantes e variáveis do modelo. A variável “N_DQMOM” representa o número de pontos de quadratura da simulação; “intphases” é o número de interações entre os pontos de quadratura, para 2, 3, 4, 5 e 6 pontos de quadratura, essa variável assume os valores 1, 3, 6, 10 e 15, respectivamente; “outlet_ID” é o número de identificação do Fluent para superfície de saída de sólidos para o *hopper*; “fluidhopper_ID” é o número de identificação do Fluent para todas as células computacionais do *hopper*; “tv_in_alpha” representa o vetor contendo a taxa volumétrica de cada ponto de quadratura na entrada do ciclone e “d_i” é o vetor que contém o diâmetro da fase sólida e dos pontos de quadratura. As variáveis “outlet_ID”, “fluidhopper_ID” e “tv_in_alpha” são utilizadas para a correção da fração volumétrica e na retirada dos sólidos que alcançam o *hopper*.

```

/*****
/*----- UDF para resolução da Equação de Balanço Populacional -----*/
/*----- Felipe Queiroz Mendes e Prof. Dr. Dirceu Noriler -----*/
*****/

/*****
/*----- Cabeçalho -----*/
#include <math.h>
#include "udf.h"
#include "linalg.h"

```

```

/*----- Definição de Constantes -----*/
#define pi 4.0*atan(1.0)
#define kv 0.523599      /*fator de forma - para esfera = pi/6 */
#define e_ss 0.9          /*coeficiente de restituição*/
#define C_fr 0.0          /*coeficiente de fricção*/
#define rho_g 1.225       /*densidade da fase gasosa*/
#define mu_g 1.7894e-5    /*viscosidade da fase sólida*/
#define rho_s 2900.0      /*densidade da fase sólida*/

int N_DQMOM = 4;         /*número de pontos de quadratura*/
int intphases = 6;       /*número interações entre quadraturas*/
int outlet_ID = 23;      /*ID da condição de contorno de saída de sólidos*/
int fluidhopper_ID = 29; /*ID do interior do hopper*/
real tv_in_alpha[] = {1.65632e-8,1.611458e-7,1.490685e-7,1.82885e-8};
/*posição 0 se refere a mistura, 1, abicissa 1, ..., N, abscissa N*/
real d_i[] = {0.0, 0.54e-6, 1.49e-6, 3.71e-6, 7.89e-6};

/*----- Subrotinas -----*/
/*cálculo da velocidade slip*/
real computeV_Slip(real V1[3], real V2[3])
{
    real V_SlipX, V_SlipY, V_SlipZ, V_Slip;

    V_SlipX = V1[0] - V2[0];
    V_SlipY = V1[1] - V2[1];
    V_SlipZ = V1[2] - V2[2];
    V_Slip = sqrt(V_SlipX*V_SlipX + V_SlipY*V_SlipY + V_SlipZ*V_SlipZ);
    return V_Slip;
}

/*-----*/
real Gidaspow_drag(real alpha_g,real V_g[3], real alpha_s, real V_s[3],
    real dSolid)
{
    real C_D, K_gs, V_SlipGS, ReS;

    V_SlipGS = computeV_Slip(V_g, V_s);
    ReS = alpha_g*rho_g*V_SlipGS*dSolid/mu_g;
    if (alpha_s < 0.2) /*modelo de wen e yu*/
    {
        if (ReS <= 1.0e-12)
        {
            C_D = 0.44;
        }
        else if(ReS <= 1000.0)
        {

```

```

        C_D = (24.0/ReS)*(1.0+0.15*pow(ReS,0.687));
    }
    else
    {
        C_D = 0.44;
    }
    K_gs=(3.0/4.0)*C_D*alpha_s*alpha_g*rho_g/dSolid*V_SlipGS*pow(
        alpha_g,-2.65);
}
else /*modelo de Ergun*/
{
    K_gs = 150.0*alpha_s*(1.0-alpha_g)*mu_g/alpha_g/dSolid/dSolid+1.75
        *rho_g*alpha_s*V_SlipGS/dSolid;
}
return K_gs;
}

/*-----*/
real Syamlal_drag(real alpha_s1, real V_s1[3], real d1, real alpha_s2,
    real V_s2[3], real d2, real soma_g0, real alpha_g)
{
    real K_ss, V_SlipSS, g0;

    V_SlipSS = computeV_Slip(V_s1, V_s2);
    g0 = 1.0/alpha_g+3.0*soma_g0*d1*d2/alpha_g/alpha_g/(d1+d2);
    K_ss=3.0*(1.0+e_ss)*(pi/2.0+C_fr*pi*pi/8.0)*alpha_s1*alpha_s2*rho_s*(
        d1+d2)*(d1+d2)*g0*V_SlipSS/(2.0*pi*(d1*d1*d1+d2*d2*d2));
    return K_ss;
}

/*-----*/
void function_v(real Vg[3], real V[3*N_DQMOM+3], real alpha_g, real
    alpha[N_DQMOM+1], real fun_V[3*N_DQMOM])
{
    real V_s1[3], V_s2[3], Kgs[N_DQMOM+1], soma_g0, Kss[N_DQMOM][N_DQMOM
        ];
    int i, j, k;
    /*coeficiente de arraste GS*/
    for (i=0; i<=N_DQMOM; i++)
    {
        V_s1[0]=V[3*i]; V_s1[1]=V[3*i+1]; V_s1[2]=V[3*i+2];
        Kgs[i]=Gidaspow_drag(alpha_g, Vg, alpha[i], V_s1, d_i[i]);
    }

    /*coeficiente de arraste SS*/
    soma_g0=0.0;
    for (i=1; i<=N_DQMOM; i++)

```

```

{
    soma_g0 += alpha[i]/d_i[i];
}
k = 0;
for (i=1; i<=N_DQMOM-1; i++)
{
    for (j=i+1; j<=N_DQMOM; j++)
    {
        V_s1[0]=V[3*i]; V_s1[1]=V[3*i+1]; V_s1[2]=V[3*i+2];
        V_s2[0]=V[3*j]; V_s2[1]=V[3*j+1]; V_s2[2]=V[3*j+2];
        Kss[i-1][j-1]=Syamlal_drag(alpha[i], V_s1, d_i[i], alpha[j],
            V_s2, d_i[j], soma_g0, alpha_g);
        Kss[j-1][i-1]=Kss[i-1][j-1];
        k = k + 1;
    }
}

/*função para calcular a velocidade*/
for (i=0; i<N_DQMOM; i++)
{
    fun_V [i] = Kgs[i+1]*(Vg[0]-V[3*i+3]) - alpha[i+1]/alpha[0]*Kgs[0]
        *(Vg[0]-V[0]);
    for (j=0; j<N_DQMOM; j++)
    {
        if (j != i)
        {
            fun_V [i] += Kss[i][j]*(V[3*j+3]-V[3*i+3]);
        }
    }
}

for (i=0; i<N_DQMOM; i++)
{
    fun_V [i+N_DQMOM] = Kgs[i+1]*(Vg[1]-V[3*i+4]) - alpha[i+1]/alpha[
        0]*Kgs[0]*(Vg[1]-V[1]);
    for (j=0; j<N_DQMOM; j++)
    {
        if (j != i)
        {
            fun_V [i+N_DQMOM] += Kss[i][j]*(V[3*j+4]-V[3*i+4]);
        }
    }
}

for (i=0; i<N_DQMOM; i++)
{

```

```

    fun_V [i+2*N_DQMOM] = Kgs[i+1]*(Vg[2]-V[3*i+5]) - alpha[i+1]/
        alpha[0]*Kgs[0]*(Vg[2]-V[2]);
    for (j=0; j<N_DQMOM; j++)
    {
        if (j != i)
        {
            fun_V [i+2*N_DQMOM] += Kss[i][j]*(V[3*j+5]-V[3*i+5]);
        }
    }
}

}

/* ----- */
void Jacobiano_v(real Vg[3], real V[3*N_DQMOM+3], real alpha_g, real
    alpha[N_DQMOM+1], real Jac_V[3*N_DQMOM][3*N_DQMOM])
{
    real delta, fd_V[3*N_DQMOM], fe_V[3*N_DQMOM];
    int i, j;

    for (i=0; i<N_DQMOM; i++)
    {
        delta=1.0e-3*V[3*i+3];
        V[3*i+3]=V[3*i+3] + delta;
        function_v(Vg, V, alpha_g, alpha, fd_V);
        V[3*i+3]=V[3*i+3] - 2.0*delta;
        function_v(Vg, V, alpha_g, alpha, fe_V);
        V[3*i+3]=V[3*i+3] + delta;
        for (j=0; j<3*N_DQMOM; j++)
        {
            if (delta == 0)
            {
                Jac_V[j][i] = 0.0;
            }
            else
            {
                Jac_V[j][i] = (fd_V[j] - fe_V[j])/(2.0*delta);
            }
        }
    }
}

for (i=0; i<N_DQMOM; i++)
{
    delta=1.0e-3*V[3*i+4];
    V[3*i+4]=V[3*i+4] + delta;
    function_v(Vg, V, alpha_g, alpha, fd_V);
    V[3*i+4]=V[3*i+4] - 2.0*delta;
    function_v(Vg, V, alpha_g, alpha, fe_V);

```



```

    V[3*i+4]=V[3*i+4] + delta;
    for (j=0;j<3*N_DQMOM;j++)
    {
        if (delta == 0)
        {
            Jac_V[j][i+N_DQMOM] = 0.0;
        }
        else
        {
            Jac_V[j][i+N_DQMOM] = (fd_V[j] - fe_V[j])/(2.0*delta);
        }
    }
}

for (i=0;i<N_DQMOM;i++)
{
    delta=1.0e-3*V[3*i+5];
    V[3*i+5]=V[3*i+5] + delta;
    function_v(Vg, V, alpha_g, alpha, fd_V);
    V[3*i+5]=V[3*i+5] - 2.0*delta;
    function_v(Vg, V, alpha_g, alpha, fe_V);
    V[3*i+5]=V[3*i+5] + delta;
    for (j=0;j<3*N_DQMOM;j++)
    {
        if (delta == 0)
        {
            Jac_V[j][i+2*N_DQMOM] = 0.0;
        }
        else
        {
            Jac_V[j][i+2*N_DQMOM] = (fd_V[j] - fe_V[j])/(2.0*delta);
        }
    }
}
}

/*-----*/
void Newton_v(real Vg[3], real V[3*N_DQMOM+3], real alpha_g, real alpha[
    N_DQMOM+1])
{
    real f_V[3*N_DQMOM], Jac_V[3*N_DQMOM][3*N_DQMOM], A1[3*N_DQMOM*3*
        N_DQMOM], B[3*N_DQMOM], X[3*N_DQMOM], max_v, max_f;
    int i, j, k, l;
    cxboolean flag;

    k=0; max_v = 1.0; max_f = 1.0; flag = 1;
    while (k < 1000 && max_v > 1.0e-6 && max_f > 1.0e-8 && flag == 1)

```

```

{
function_v(Vg, V, alpha_g, alpha, f_V);
Jacobiano_v(Vg, V, alpha_g, alpha, Jac_V);

/*matriz A*/
for (i=0; i<3*N_DQMOM; i++)
{
    for (j=0; j<3*N_DQMOM; j++)
    {
        A1[i*3*N_DQMOM+j]=Jac_V[i][j];
        if (i == j && Jac_V[i][j] == 0.0)
        {
            A1[i*3*N_DQMOM+j]=1.0;
        }
    }
}

/*vetor B*/
for (i=0; i<3*N_DQMOM; i++)
{
    B[i] = -f_V[i];
}

/*Eliminação de Gauss*/
flag = Gauss_Elimination_dp(3*N_DQMOM, 1, A1, X, B, 1);
l=0;
max_v=0.0;
max_f=0.0;
for (i=0; i<3; i++)
{
    for (j=0; j<N_DQMOM; j++)
    {
        V[3*j+i+3]=V[3*j+i+3]+X[l];
        if (fabs(V[3*j+i+3])>fabs(Vg[i]))
        {
            V[3*j+i+3] = V[i];
            flag = 0;
        }
        if (max_v<fabs(X[l]/V[3*j+i+3]) && V[3*j+i+3] != 0.0)
        {
            max_v=fabs(X[l]/V[3*j+i+3]);
        }
        if (max_f<fabs(f_V[l]))
        {
            max_f=fabs(f_V[l]);
        }
    }
    l=l+1;
}

```

```

    }
  }
  k=k+1;
}
}

/*****
/*----- Cálculo das velocidades para cada ponto de quadratura -----*/

DEFINE_ADJUST(quadrature_point_velocity,d)
{
  Thread *tg, *ts, *t;
  cell_t c;
  real Vg[3], V[3*N_DQMOM+3];
  real alpha_g, alpha[N_DQMOM+1];
  real momento2, momento3;
  int i;

  /*Retirar valores negativos das abscissas*/
  thread_loop_c(t,d)
  {
    if (FLUID_THREAD_P(t))
    {
      tg = THREAD_SUB_THREAD(t,0); /*obter thread gás*/
      begin_c_loop(c,t)
      {
        for (i=0; i<N_DQMOM; i++)
        {
          if (C_UDSI(c,tg,i) <= 0.0)
          {
            C_UDSI(c,tg,i) = 0.0;
          }
        }
      }
      end_c_loop(c,t)
    }
  }

  thread_loop_c(t,d)
  {
    tg = THREAD_SUB_THREAD(t,0); /*obter thread gás*/
    ts = THREAD_SUB_THREAD(t,1); /*obter thread sólido*/
    begin_c_loop(c,t)
    {
      /*velocidades*/
      Vg[0] = C_U(c,tg); Vg[1] = C_V(c,tg); Vg[2] = C_W(c,tg);
      V[0] = C_U(c,ts); V[1] = C_V(c,ts); V[2] = C_W(c,ts);
    }
  }
}

```

```

for (i=0; i<3*N_DQMOM; i++)
{
    V[i+3] = C_UDMI(c,t,i);
}

/*fração volumétrica*/
alpha_g = C_VOF(c,tg); alpha[0] = C_VOF(c,ts);
for (i=0; i<N_DQMOM; i++)
{
    alpha[i+1] = C_UDSI(c,tg,i);
}

/*início do cálculo das velocidades*/
if (alpha[0] <= 5.0e-8)
{
    for (i=0; i<3*N_DQMOM; i++)
    {
        C_UDMI(c,t,i) = 0.0;
        C_UDMI(c,t,3*N_DQMOM) = d_i[1];
    }
}
else
{
    /*início diâmetro da fase sólida*/
    momento2 = 0.0;
    momento3 = 0.0;
    for (i=1; i<=N_DQMOM; i++)
    {
        momento2 += alpha[i] / kv / d_i[i];
        momento3 += alpha[i] / kv;
    }
    if (momento2 <= 1.0e-10)
    {
        d_i[0] = d_i[1];
    }
    else
    {
        d_i[0] = momento3/momento2;
    }
    C_UDMI(c,t,3*N_DQMOM)=d_i[0];
    /*final diâmetro da fase sólida*/

    /*método de Newton para resolução de sistema não-linear*/
    Newton_v(Vg, V, alpha_g, alpha);
    for (i=0; i<3*N_DQMOM; i++)
    {
        C_UDMI(c,t,i) = V[i+3];/*armazenamento das velocidades*/
    }
}

```

```

        }
    }
    /*final do cálculo das velocidades*/
}
end_c_loop(c,t)
}

}

/*****
/*----- Define o diâmetro da partícula -----*/

DEFINE_PROPERTY(diameter_particle, c, t)
{
    real diameter;
    diameter = C_UDMI(c,t,3*N_DQMOM);
    return diameter;
}

/*****
/*----- Define o termo transiente -----*/

DEFINE_UDS_UNSTEADY(my_uds_unsteady_PBE,c,t,i,apu,su)
{
    real physical_dt, vol, phi_old;
    physical_dt = RP_Get_Real("physical-time-step");
    vol = C_VOLUME(c,t);
    *apu = -rho_s * vol / physical_dt; /* parte implícita */
    phi_old = C_UDSI_M1(c,t,i);
    *su = rho_s*vol * phi_old / physical_dt; /* parte explícita */
}

/*****
/*----- Define o termo convectivo -----*/

DEFINE_UDS_FLUX(my_uds_flux_PBE,f,t,i)
{
    cell_t c0, c1 = -1;
    Thread *t0, *t1 = NULL;
    real NV_VEC(psi_vec), NV_VEC(A), flux = 0.0;
    c0 = F_C0(f,t);
    t0 = F_C0_THREAD(f,t);
    F_AREA(A, f, t);
    /* Se pertencer a uma condição de contorno, usar valores da face; */
    /* caso contrário, usar média das células adjacentes. */
    if (BOUNDARY_FACE_THREAD_P(t))

```

```

{
    NV_DS(psi_vec, =, F_UDMI(f,t,3*i), F_UDMI(f,t,3*i+1), F_UDMI(f,t,3*i
        +2), *,rho_s);
    flux = NV_DOT(psi_vec, A); /* fluxo atraves da face */
}
else
{
    c1 = F_C1(f,t); /* célula vizinha a face */
    t1 = F_C1_THREAD(f,t);
    NV_DS(psi_vec, =, C_UDMI(c0,t0,3*i),C_UDMI(c0,t0,3*i+1),C_UDMI(c0,t0
        ,3*i+2),*,rho_s);
    NV_DS(psi_vec, +=, C_UDMI(c1,t1,3*i),C_UDMI(c1,t1,3*i+1),C_UDMI(c1,
        t1,3*i+2),*,rho_s);
    flux = NV_DOT(psi_vec, A)/2.0; /* fluxo médio através da face */
}
return flux;
}

/*****
/*----- Correção da fração volumétrica -----*/

DEFINE_EXECUTE_AT_END(volume_fraction_correction)
{
    Domain *d;
    Thread *t, *tout, *thopper, *ts, *tg, *t0;
    cell_t c, c0;
    face_t f;
    int i, j;
    real volume_c, volume[N_DQMOM], volume_s, volume_new[N_DQMOM],
        volume_old[N_DQMOM], difvol[N_DQMOM], maior, sumvol;
    real alpha[N_DQMOM], alpha_s, sum_alpha, alpha_new[N_DQMOM];
    real time_step, NV_VEC(psi_vec), NV_VEC(A), tv_out_s,
        tv_out_alpha_sum,
        tv_out_alpha[N_DQMOM];
    real volume_s_hopper, volume_hopper[N_DQMOM], volume_hopper_sum;

    d = Get_Domain(1); /* domínio da mistura */

    /*Retirar valores negativos das abscissas*/
    thread_loop_c(t,d)
    {
        if (FLUID_THREAD_P(t))
        {
            tg = THREAD_SUB_THREAD(t,0); /*obter thread do gás*/
            begin_c_loop(c,t)
            {
                for (i=0; i<N_DQMOM; i++)

```

```

        {
            if (C_UDSI(c,tg,i) <= 0.0)
            {
                C_UDSI(c,tg,i) = 0.0;
            }
        }
    }
    end_c_loop(c,t)
}

/*início balanço global para manter o volume constante*/
/*volume no tempo anterior*/
for (i=0; i<N_DQMOM; i++)
{
    volume[i] = 0.0;
}
thread_loop_c(t,d)
{
    if (FLUID_THREAD_P(t))
    {
        tg = THREAD_SUB_THREAD(t,0); /*obter thread gas*/
        begin_c_loop(c,t)
        {
            volume_c = C_VOLUME(c,t); /*volume de cada célula*/
            for (i=0; i<N_DQMOM; i++)
            {
                /*fração volumétrica da quadratura no passo de tempo anterior*/
                alpha[i] = C_UDSI_M1(c,tg,i);
                /*volume total da quadratura no passo de tempo anterior*/
                volume[i] += alpha[i] * volume_c;
            }
        }
        end_c_loop(c,t)
    }
}

/* soma sobre todas as partições*/
for (i=0; i<N_DQMOM; i++)
{
    volume[i] = PRF_GRSUM1(volume[i]);
}

/*BC inlet_ID*/
time_step = RP_Get_Real("physical-time-step");
for (i=0; i<N_DQMOM; i++)
{
    volume[i] += tv_in_alpha[i]*time_step;
}

```

```

}

/*BC outlet_air*/
tout = Lookup_Thread(d, outlet_ID);
tg = THREAD_SUB_THREAD(tout,0);
ts = THREAD_SUB_THREAD(tout,1);
tv_out_s = 0.0;
for (i=0;i<N_DQMOM;i++)
{
    tv_out_alpha[i] = 0.0;
}
begin_f_loop(f,tout)
{
    tv_out_s += F_FLUX(f,ts) / rho_s; /*taxa massica da fase solida*/
    F_AREA(A, f, tout); /*área da face*/
    c0=F_CO(f,tg);t0=F_CO_THREAD(f,tg);/*Thread e cell dona da face*/
    if (F_FLUX(f,ts) > 0.0)
    {
        for (i=0;i<N_DQMOM;i++)
        {
            NV_DS(psi_vec, =, C_UDMI(c0,t0,3*i), C_UDMI(c0,t0,3*i+1),
                C_UDMI(c0,t0,3*i+2),*, rho_s);
            /*taxa massica da quadratura*/
            tv_out_alpha[i]+=C_UDSI(c0,t0,i)*NV_DOT(psi_vec, A)/rho_s;
        }
    }
}
end_f_loop(f, tout)
/*soma sobre todas as partições*/
tv_out_s = PRF_GRSUM1(tv_out_s);
for (i=0;i<N_DQMOM;i++)
{
    tv_out_alpha[i] = PRF_GRSUM1(tv_out_alpha[i]);
}

tv_out_aplha_sum = 0.0;
for (i=0;i<N_DQMOM;i++)
{
    tv_out_aplha_sum += tv_out_alpha[i];
}
if (tv_out_aplha_sum < 1.0e-12 || tv_out_s < 1.0e-12)
{
    for (i=0;i<N_DQMOM;i++)
    {
        tv_out_alpha[i] = 0.0;
    }
}
}

```



```

else
{
    for (i=0;i<N_DQMOM;i++)
    {
        tv_out_alpha[i] = tv_out_alpha[i]*tv_out_s/tv_out_aplha_sum;
    }
}
for (i=0;i<N_DQMOM;i++)
{
    volume[i] -= tv_out_alpha[i]*time_step;
}

/*BC outlet_hopper*/
thopper = Lookup_Thread(d, fluidhopper_ID);
tg = THREAD_SUB_THREAD(thopper,0);
ts = THREAD_SUB_THREAD(thopper,1);
volume_s_hopper = 0.0;
for (i=0;i<N_DQMOM;i++)
{
    volume_hopper[i] = 0.0;
}
begin_c_loop(c,thopper)
{
    volume_c = C_VOLUME(c,thopper); /*volume de cada célula*/
    alpha_s = C_VOF(c,ts); /*fração volumétrica do sólido*/
    volume_s_hopper += alpha_s * volume_c; /*volume total do sólido*/
    C_VOF(c,tg) = 1.0;
    C_VOF(c,ts) = 0.0;
    for (i=0; i<N_DQMOM; i++)
    {
        /*fração volumétrica da quadratura no hopper*/
        alpha[i] = C_UDSI(c,tg,i);
        /*volume total da quadratura no hopper*/
        volume_hopper[i] += alpha[i] * volume_c;
        C_UDSI(c,tg,i) = 0.0;
    }
}
end_c_loop(c,thopper)
/*soma sobre todas as partições*/
volume_s_hopper = PRF_GRSUM1(volume_s_hopper);
for (i=0;i<N_DQMOM;i++)
{
    volume_hopper[i] = PRF_GRSUM1(volume_hopper[i]);
}
/*soma de todas as quadraturas*/
volume_hopper_sum = 0.0;
for (i=0;i<N_DQMOM;i++)

```

```

{
    volume_hopper_sum += volume_hopper[i];
}
if (volume_hopper_sum < 1.0e-12 || volume_s_hopper < 1.0e-12)
{
    for (i=0;i<N_DQMOM;i++)
    {
        volume_hopper[i] = 0.0;
    }
}
else
{
    for (i=0;i<N_DQMOM;i++)
    {
        volume_hopper[i] = volume_hopper[i]*volume_s_hopper/
            volume_hopper_sum;
    }
}
for (i=0;i<N_DQMOM;i++)
{
    volume[i] -= volume_hopper[i];
}

/*volume de sólidos no ciclone*/
volume_s = 0.0;
thread_loop_c(t,d)
{
    if (FLUID_THREAD_P(t))
    {
        ts = THREAD_SUB_THREAD(t,1); /*obter thread solido*/
        begin_c_loop(c,t)
        {
            volume_c = C_VOLUME(c,t); /*volume de cada célula*/
            alpha_s = C_VOF(c,ts);    /*fração volumétrica do sólido*/
            volume_s += alpha_s * volume_c; /*volume total do sólido*/
        }
        end_c_loop(c,t)
    }
}
volume_s = PRF_GRSUM1(volume_s); /* soma sobre todas as partições*/
/*soma dos volumes*/
sumvol = 0.0;
for (i=0; i<N_DQMOM; i++)
{
    sumvol += volume[i];
}
/*correção dos volumes*/

```

```

for (i=0; i<N_DQMOM; i++)
{
    volume[i] = volume[i]*volume_s/sumvol;
    volume_new[i] = 0.0;
}
/*final do balanço global para manter o volume constante*/

/*início da correção da fração volumétrica*/
thread_loop_c(t,d)
{
    if (FLUID_THREAD_P(t))
    {
        tg = THREAD_SUB_THREAD(t,0); /*obter thread gás*/
        ts = THREAD_SUB_THREAD(t,1); /*obter thread sólido*/
        begin_c_loop(c,t)
        {
            volume_c = C_VOLUME(c,t); /*volume de cada célula*/
            alpha_s = C_VOF(c,ts); /*fração volumétrica do sólido*/
            sum_alpha = 0.0;
            for (i=0; i<N_DQMOM; i++)
            {
                alpha[i] = C_UDSI(c,tg,i)*volume_s/sumvol;
                sum_alpha += alpha[i]; /*soma que deveria ser alpha_s*/
            }
            if (sum_alpha > 1.0e-16)
            {
                /*normalização da fração volumétrica*/
                for (i=0; i<N_DQMOM; i++)
                {
                    alpha_new[i] = alpha[i]*alpha_s/sum_alpha;
                    volume_new[i] += alpha_new[i] * volume_c;
                    C_UDSI(c,tg,i) = alpha_new[i]; /*novo alpha[i]*/
                }
            }
        }
        end_c_loop(c,t)
    }
}
/* soma sobre todas as partições*/
for (i=0; i<N_DQMOM; i++)
{
    volume_new[i] = PRF_GRSUM1(volume_new[i]);
}

/*diferença entre os volumes das abscissas*/
for (i=0; i<N_DQMOM; i++)

```

```

{
    difvol[i] = fabs(volume_new[i]-volume[i])/volume[i];
    volume_old[i] = volume_new[i];
}
maior = difvol[0];
for(i=1; i<N_DQMOM;i++)
{
    if(maior<difvol[i])
    {
        maior=difvol[i];
    }
}

/*processo iterativo*/
j=0;
while(maior > 1.0e-6 && j < 10)
{
    for (i=0; i<N_DQMOM; i++)
    {
        volume_new[i] = 0.0;
    }
    thread_loop_c(t,d)
    {
        if (FLUID_THREAD_P(t))
        {
            tg = THREAD_SUB_THREAD(t,0); /*obter thread gás*/
            ts = THREAD_SUB_THREAD(t,1); /*obter thread sólido*/
            begin_c_loop(c,t)
            {
                volume_c = C_VOLUME(c,t); /*volume de cada célula*/
                alpha_s = C_VOF(c,ts); /*fração volumétrica do sólido*/
                sum_alpha = 0.0;
                for (i=0; i<N_DQMOM; i++)
                {
                    /*correção da fração volumétrica das abscissas*/
                    alpha[i] = C_UDSI(c,tg,i)*volume[i]/volume_old[i];
                    sum_alpha+=alpha[i]; /*soma que deveria ser alpha_s*/
                }
                if (sum_alpha > 1.0e-16)
                {
                    /*normalização da fração volumétrica*/
                    for (i=0; i<N_DQMOM; i++)
                    {
                        alpha_new[i] = alpha[i]*alpha_s/sum_alpha;
                        volume_new[i] += alpha_new[i] * volume_c;
                        C_UDSI(c,tg,i) = alpha_new[i]; /*novo alpha[i]*/
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    end_c_loop(c,t)
}
}
/* soma sobre todas as partições*/
for (i=0; i<N_DQMOM; i++)
{
    volume_new[i] = PRF_GRSUM1(volume_new[i]);
}

/*diferença entre os volumes das abscissas*/
for (i=0; i<N_DQMOM; i++)
{
    difvol[i] = fabs(volume_new[i]-volume[i])/volume[i];
    volume_old[i] = volume_new[i];
}

maior = difvol[0];
for(i=1; i<N_DQMOM;i++)
{
    if(maior<difvol[i])
    {
        maior=difvol[i];
    }
}
j++;
}

/*correção dos valores nas faces*/
thread_loop_f(t,d)
{
    tg = THREAD_SUB_THREAD(t,0); /*obter thread do gás*/
    begin_f_loop(f,t)
    {
        if (BOUNDARY_FACE_THREAD_P(t))
        {
            c0 = F_CO(f,tg); t0 = F_CO_THREAD(f,tg);
            for (i=0;i<N_DQMOM;i++)
            {
                F_UDSI(f,tg,i)= C_UDSI(c0,t0,i);
            }
        }
    }
    end_f_loop(f, t)
}
}

```

APÊNDICE B – CONDIÇÕES DE CONTORNO

Este apêndice contém as condições de contorno que precisaram ser implementadas para simulação tanto do leito fluidizado circulante quanto do ciclone com alta carga.

B.1 PERFIL DE VELOCIDADE

A entrada de gás no CFB é dada por um perfil de velocidade de modo a representar oito bicos injetores. O código para essa condição de contorno está escrito abaixo:

```

DEFINE_PROFILE(inlet_gas_velocity, t, i)
{
    real y[ND_ND];
    real x;
    face_t f;
    begin_f_loop(f,t)
    {
        F_CENTROID(y,f,t);
        x = y[0];
        if (x <= 0.015)
        {
            F_PROFILE(f,t,i) = 0;
        }
        else if( x > 0.015 && x <= 0.025)
        {
            F_PROFILE(f,t,i) = 1125.*x-16.875;
        }
        else if( x > 0.025 && x <= 0.035)
        {
            F_PROFILE(f,t,i) = 39.375-1125.*x;
        }
        else if( x > 0.035 && x <= 0.065)
        {
            F_PROFILE(f,t,i) = 0;
        }
        else if( x > 0.065 && x <= 0.075)
        {
            F_PROFILE(f,t,i) = 1125.*x-73.125;
        }
    }
}

```

```

else if( x > 0.075 && x <= 0.085)
{
    F_PROFILE(f,t,i) = 95.625-1125.*x;
}
else if( x > 0.085 && x <= 0.115)
{
    F_PROFILE(f,t,i) = 0;
}
else if( x > 0.115 && x <= 0.125)
{
    F_PROFILE(f,t,i) = 1125.*x-129.375;
}
else if( x > 0.125 && x <= 0.135)
{
    F_PROFILE(f,t,i) = 151.875-1125.*x;
}
else if( x > 0.135 && x <= 0.165)
{
    F_PROFILE(f,t,i) = 0;
}
else if( x > 0.165 && x <= 0.175)
{
    F_PROFILE(f,t,i) = 1125.*x-185.625;
}
else if( x > 0.175 && x <= 0.185)
{
    F_PROFILE(f,t,i) = 208.125-1125.*x;
}
else if( x > 0.185 && x <= 0.215)
{
    F_PROFILE(f,t,i) = 0;
}
else if( x > 0.215 && x <= 0.225)
{
    F_PROFILE(f,t,i) = 1125.*x-241.875;
}
else if( x > 0.225 && x <= 0.235)
{
    F_PROFILE(f,t,i) = 264.375-1125.*x;
}
else if( x > 0.235 && x <= 0.265)
{
    F_PROFILE(f,t,i) = 0;
}
else if( x > 0.265 && x <= 0.275)
{
    F_PROFILE(f,t,i) = 1125.*x-298.125;
}

```

```

    }
    else if( x > 0.275 && x <= 0.285)
    {
        F_PROFILE(f,t,i) = 320.625-1125.*x;
    }
    else if( x > 0.285 && x <= 0.315)
    {
        F_PROFILE(f,t,i) = 0;
    }
    else if( x > 0.315 && x <= 0.325)
    {
        F_PROFILE(f,t,i) = 1125.*x-354.375;
    }
    else if( x > 0.325 && x <= 0.335)
    {
        F_PROFILE(f,t,i) = 376.875-1125.*x;
    }
    else if( x > 0.335 && x <= 0.365)
    {
        F_PROFILE(f,t,i) = 0;
    }
    else if( x > 0.365 && x <= 0.375)
    {
        F_PROFILE(f,t,i) = 1125.*x-410.625;
    }
    else if( x > 0.375 && x <= 0.385)
    {
        F_PROFILE(f,t,i) = 433.125-1125.*x;
    }
    else
    {
        F_PROFILE(f,t,i) = 0;
    }
}
end_f_loop(f, t)
}

```

B.2 ENTRADA DE SÓLIDOS

Em um leito fluidizado circulante há a recirculação de sólidos, sendo assim, todas as partículas que deixam o leito deve retornar a este, inclusive as frações volumétricas para cada ponto de quadratura. Assim, usando a sub-rotina *DEFINE_ADJUST* foi possível calcular o fluxo de sólidos que saem do CFB e utilizando a sub-rotina *DEFINE_PROFILE* foi implementada a condição de contorno de fluxo de entrada de sólidos no Fluent.

O código foi descrito abaixo e nele deve-se modificar a dimensão do vetor “talpa” dependendo do número de pontos de quadraturas utilizados.

```

real mass_flow_rate, talpa[4];

DEFINE_ADJUST(compute_solid_mass_flow, domain)
{
    Domain *d;
    Thread *t;
    face_t f;
    cell_t c0;
    Thread *t0;
    int i;
    real NV_VEC(psi_vec), NV_VEC(A), taplha_sum;
    d = Get_Domain(3); /*obter domínio sólido ID=3*/
    t = Lookup_Thread(d, 15); /*obter Thread da BC outlet ID=15*/

    mass_flow_rate = 0.0;
    for (i=0; i<4; i++)
    {
        talpa[i] = 0.0;
    }
    begin_f_loop(f, t)
    {
        mass_flow_rate += F_FLUX(f, t);
        F_AREA(A, f, t);
        c0 = F_CO(f, t); t0 = F_CO_THREAD(f, t);
        if (F_FLUX(f, t) > 0.0)
        {
            for (i=0; i<4; i++)
            {
                NV_DS(psi_vec, =, C_UDMI(c0, t0, 3*i), C_UDMI(c0, t0, 3*i+1),
                    C_UDMI(c0, t0, 3*i+2), *, C_R(c0, t0));
                talpa[i] += C_UDSI(c0, t0, i) * NV_DOT(psi_vec, A);
            }
        }
    }
    end_f_loop(f, t)

    /*soma sobre todas as partições*/
    mass_flow_rate = PRF_GRSUM1(mass_flow_rate);
    for (i=0; i<4; i++)
    {
        talpa[i] = PRF_GRSUM1(talpa[i]);
    }
    taplha_sum = 0.0;
    for (i=0; i<4; i++)

```

```

{
    taplha_sum += talpha[i];
}

if (taplha_sum < 1.0e-12 || mass_flow_rate < 1.0e-12)
{
    for (i=0;i<4;i++)
    {
        talpha[i]=0.0;
    }
}
else
{
    for (i=0;i<4;i++)
    {
        talpha[i]=talpha[i]*mass_flow_rate/taplha_sum;
    }
}
}

DEFINE_PROFILE(Inlet_mass_flow_rate_solid, t, i)
{
    face_t f;

    begin_f_loop(f,t)
    {
        F_PROFILE(f,t,i) = mass_flow_rate;
    }
    end_f_loop(f, t)
}

DEFINE_PROFILE(inlet_mass_flow_rate_gas, t, i)
{
    face_t f;
    real rho_s = 2480.0;
    real rho_g = 1.225;
    begin_f_loop(f,t)
    {
        F_PROFILE(f,t,i) = mass_flow_rate*rho_g/rho_s;
    }
    end_f_loop(f, t)
}

DEFINE_PROFILE(inlet_alpha, t, i)
{
    face_t f;

```

```
begin_f_loop(f,t)
{
    F_PROFILE(f,t,i) = talpha[i];
}
end_f_loop(f, t)
}
```

B.3 SAÍDA DOS PONTOS DE QUADRATURA

Foi considera na saída dos pontos de quadratura que a fração volumétrica do centro da célula computacional seria o valor que deixaria a fronteira do sistema. Assim sendo, o código implementado foi escrito como:

```
DEFINE_PROFILE(outlet_scalars, t, i)
{
    face_t f;
    cell_t c0;
    Thread *t0;

    begin_f_loop(f,t)
    {
        c0 = F_C0(f,t); t0 = F_C0_THREAD(f,t);
        if (F_FLUX(f,t) <= 0.0)
        {
            F_PROFILE(f,t,i) = 0.0;
        }
        else
        {
            F_PROFILE(f,t,i) = C_UDSI(c0,t0,i);
        }
    }
    end_f_loop(f, t)
}
```

APÊNDICE C – ALGORITMO PRODUTO-DIFERENÇA

O algoritmo Produto-Diferença foi escrito em Fortran e a partir de $2N$ momentos ele calcula N pesos e abscissas. Deste modo, é necessário alterar o vetor “m” com os momentos da distribuição normalizada e o escalar “N” com o número de pontos de quadraturas desejado. É importante ressaltar ainda que este programa necessita de uma sub-rotina para calcular os autovalores e autovetores que se encontra no Anexo B.

```
! Algoritmo Produto-Diferenca
! Determina N pesos e abscissas a partir de 2N momentos
!
      program pd

      real*8 m, xi, w
      integer it_max, N, i
      dimension m(12),xi(6),w(6)
      data m/1.d0,2.56822427d0,8.4615887d0,33.145436d0,149.46495555d0,
+       764.496586d0,4398.3598d0,28199.069d0,198674.06d0,1510887.8d0,
+       3.34761d-4, 2.206142d-4/
      data N, it_max /4, 100/

      call algopd(m,N,it_max, xi, w)

      open(unit=22,file='saida.txt')
      write(22,10) (i,w(i),i,xi(i), i=1,N)
10    format(1x,'Peso ',i1,' =',f9.5,'  Abixissa ',i1,' =',f9.5/)

      stop
      end

!
! *****
! Sub-rotina algoritmo Produto-Diferenca.
!
      subroutine algopd(m,N,it_max, xi, w)
      real*8 P, beta, JAC, VEC, lambda, m, xi, w
      integer i, j, it_num, rot_num, N, it_max
      dimension P(2*N+1,2*N+1), beta(2*N), JAC(N,N), VEC(N,N),
+       lambda(N), xi(N), w(N), m(2*N)
! matriz P
! matriz nula
      do 11 i=1,2*N+1
```

```

        do 10 j=1,2*N+1
            P(i,j) = 0.d0
10         continue
11         continue
! primeira coluna de P
        P(1,1) = 1.d0
! segunda coluna de P
        do 12 i=1,2*N
            P(i,2) = (-1)**(i+1)*m(i)
12         continue
! demais colunas de P
        do 14 j=3,2*N+1
            do 13 i = 1,2*N+2-j
                P(i,j)=P(1,j-1)*P(i+1,j-2)-P(1,j-2)*P(i+1,j-1)
13             continue
14         continue
! vetor beta
        beta(1) = 0.d0
        do 15 j=2,2*N
            beta(j)=P(1,j+1) / P(1,j) / P(1,j-1)
15         continue
! matriz Jac
        do 17 i=1,N
            do 16 j=1,N
                JAC(i,j) = 0.d0
                if(i.eq.j) then
                    JAC(i,j) = beta(2*j)+beta(2*j-1)
                endif
                if(i+1.eq.j) then
                    JAC(i,j) = -(beta(2*(j-1))*beta(2*(j-1)+1))**(0.5d0)
                endif
                if(i.eq.j+1) then
                    JAC(i,j) = -(beta(2*j)*beta(2*j+1))**(0.5d0)
                endif
16             continue
17         continue
! obter autovalores e autovetores
        call jacobi_eigenvalue (N, JAC, it_max, VEC, lambda, it_num,
&         rot_num)
! retornar pesos e abcissas
        do 18 i=1,N
            xi(i) = lambda(i)
            w(i) = m(1)*VEC(1,i)*VEC(1,i)
18         continue

        return
    end

```

Anexos

ANEXO A – MÉTODOS DE RECONSTRUÇÃO DA PDF

Neste estudo foram avaliados os métodos de máxima entropia e *spline* cúbica para reconstrução da PDF a partir de seus momentos. O código destes métodos estão expostos nos próximas seções.

É interessante salientar que para caso haja dificuldades para convergência de ambos os métodos pode-se mudar a magnitude dos valores dos momentos alterando as unidades destes, ou seja, considerando que a unidade do momento de primeira ordem seja μm pode ser que convertendo esse momento para mm melhore a convergência dos métodos.

A.1 MÁXIMA ENTROPIA

O método de máxima entropia possui poucas variáveis para ser configuradas. É necessário fornecer o intervalo que a função está contida preenchendo as variáveis “xmin” e “xmax” e também o intervalo entre os pontos da distribuição, “dx”. E, por fim, basta adicionar o número de momentos desejados no vetor “mu” para se obter a reconstrução da PDF.

```
clear all; clf;
% Maxima Entropia

xmin=0.0; xmax=8.0; dx=1.0e-2; % define the x axis
x=[xmin:dx:xmax]';

mu(1) = 1.00;
mu(2) = 2.64055e+02*1.0e-2;
mu(3) = 8.11618e+04*1.0e-4;
mu(4) = 2.88274e+07*1.0e-6;
mu(5) = 1.16942e+10*1.0e-8;
mu(6) = 5.33360906514972e+12*1.0e-10;
mu(7) = 2.68615963646757e+15*1.0e-12;
mu(8) = 1.46711536956573e+018*1.0e-14;
mu(9) = 8.54941312261519e+020*1.0e-16;
mu(10) = 5.24334002231077e+023*1.0e-18;

tic
mu=mu';
[lambda,p,entr]=me_dens2(mu,x);
```

```

plot(x,p);
toc
s=[100*x,p/100];
save ME10.mat s

function [lambda,p,entr]=me_dens2(mu,x,lambda0)
    %ME_DENS2
    % [LAMBDA,P,ENTR]=ME_DENS2(MU,X,LAMBDA0)
    % This program calculates the Lagrange Multipliers of the ME
    % probability density functions p(x) from the knowledge of the
    % N moment constraints in the form:
    %  $E(x^n)=\mu(n)$   $n=0:N$  with  $\mu(0)=1$ .
    %
    % MU is a table containing the constraints MU(n),n=1:N.
    % X is a table defining the range of the variation of x.
    % LAMBDA0 is a table containing the first estimate of the LAMBDA's.
    % (This argument is optional.)
    % LAMBDA is a table containing the resulting Lagrange parameters.
    % P is a table containing the resulting pdf p(x).
    % ENTR is a table containing the entropy values at each
    % iteration.
    %
    % Author: A. Mohammad-Djafari
    % Date : 10-01-1991
    %
    x=x(:); lx=length(x); % x axis
    xmin=x(1); xmax=x(lx); dx=x(2)-x(1);
    %
    if(nargin == 2) % initialize LAMBDA
        lambda=zeros(size(mu)); % This produces a uniform
        lambda(1)=log(xmax-xmin); % distribution.
    else
        lambda= lambda0 (:);
    end
    N=length(lambda);
    %
    M=2*N-1;
    fin=zeros(length(x),M); %
    fin(:,1)=ones(size(x)); % fi0(x)=1
    for n=2:M
        fin(:,n)=x.*fin(:,n-1);
    end
    %
    iter=0;
    while 1 % start iterations
        iter=iter+1;
        disp('-----'); disp(['iter=',num2str(iter)]);
        %

```



```

p=exp(-(fin(:,1:N)*lambda)); % Calculate p(x)
plot(x,p); % plot it
%
G=zeros(M,1); % Calculate Gn
for n=1:M
    G(n)=dx*sum(fin(:,n).*p);
end
%
entr(iter)=lambda'*G(1:N); % Calculate the entropy value
disp(['Entropy=',num2str(entr(iter))])
%
gnk=zeros(N,N); % Calculate gnk
for i=1:N % Matrix G is a Hankel matrix
    gnk(:,i)=-G(i:N+i-1);
end
%
v=mu-G(1:N); % Calculate v
delta=gnk\v; % Calculate delta
lambda=lambda+delta; % Calculate lambda
eps=1e-6; % Stopping rules
if(abs(delta./lambda)<eps), break, end
if(iter>2)
    if(abs((entr(iter)-entr(iter-1))/entr(iter))<eps),break, end
end
end
%
p=exp(-(fin(:,1:N)*lambda)); % Calculate the final p(x)
plot(x,p); % plot it
entr=entr(:);
disp('----- END -----')
end

```

A.2 *SPLINE* CÚBICA

O método da *spline* cúbica possui mais variáveis para ser configuradas. Além do intervalo que contém a variável, “a” e “ori_length”, e do vetor de momentos, “mu”, mais seis variáveis necessitam ser configuradas. A primeira, “k”, representa o número total de momentos que serão utilizados na reconstrução; a variável “exemple” possui o nome do arquivo contém o vetor de momentos; “minimum” possui o valor do maior valor negativo permitido na reconstrução; “left_second_der” permite alterar a condição de contorno da esquerda da distribuição; “min_space” é o fator que permite um menor espaçamento entre nós e, finalmente, “b_domain” que permite alargar o domínio da distribuição. Dependendo dos valores utilizados para “minimum”, “left_second_der”, “min_space” e “b_domain” pode-se chegar numa reconstrução diferente o que torna esse método mais difícil de ser

utilizado.

```
% Matlab script to reconstruct a distribution from its moments
% using cubic splines. Original algorithm proposed by John et al. (2007) and
% modified by Luis G. M. de Souza, september 2008.

% Parameters
% a -> left boundary of interval for reconstruction, set to zero below
% ori_length -> initial length of the interval
% k -> number of moments
% example.m -> contains the moments of the function
% reference.m -> contains the reference distribution data (Not
% needed to run the script)
% minimum -> allowed relative negative values
% left_second_der -> if not zero, second derivative at left boudary used
% min_space -> minimum space factor allowed between two nodes. Used as (bmin_space)
% b_domain -> domain for the non equidistant redistribution of
% the nodes. To make the domain a little bit bigger then the one
% found in the equidistant grid

clear all
clc

% To run the reconstruction, leave the parameters below uncommented.
a=0;
ori_length=15.0;
k=10;
example='moments';
% reference='reference.m';
minimum=0;
left_second_der=1;
min_space=10;
b_domain=1.2;

clf
tic
if k<0
    error('k too small')
end

% set interval for computations to [0,1]
a = 0;
b = 1;
scale = b/ori_length;

% Cubic spline
s=3;
```

```
% Number of intervals "n"
n = k+s;

% weighting factor
omega = 1e-4;

% define arrays
A=zeros((s+1)*n,(s+1)*n);
rhs=zeros((s+1)*n,1);
x=zeros((s+1)*n,1);

% Number of iterations
number = 0;

% compute nodes for equidistant grid
h = (b-a)/n;
for i=1:n+1
    x(i) = a + h*(i-1);
end
x(n+1) = b;

%
% Flag for adaptive grid
auto_regrid = 1;

while 1

% loop over redistribution of the points
while 1

    % Set flags
    redistribute_left = 0;
    redistribute_right = 0;
    redistribute_nodes = 0;

    % fill matrix with the conditions for the cubic spline
    % left boudnary
    A(1,1) = 1;

    % these conditions have to be included if 2nd derivative at
    % left boundary is zero
    if left_second_der == 1
        A(2,2) = 1;
        A(3,3) = 2;
    end

    % Linear spline is used in first interval
```

```

if left_second_der == 0
    A(2,3) = 1;
    A(3,4) = 2;
end
if left_second_der == 2
    A(2,2) = 1;
    A(3,4) = 2;
end

% Interior nodes
for j=1:n-1
    A(3*j+1,4*j-3) = 1;
    A(3*j+1,4*j-2) = x(j+1) - x(j);
    A(3*j+1,4*j-1) = (x(j+1) - x(j))^2;
    A(3*j+1,4*j) = (x(j+1) - x(j))^3;
    A(3*j+1,4*j+1) = -1;
    A(3*j+2,4*j-2) = 1;
    A(3*j+2,4*j-1) = 2*(x(j+1) - x(j));
    A(3*j+2,4*j) = 3*(x(j+1) - x(j))^2;
    A(3*j+2,4*j+1) = -1;
    A(3*j+3,4*j-1) = 2;
    A(3*j+3,4*j) = 6*(x(j+1) - x(j));
    A(3*j+3,4*j+1) = -2;
end

% right boundary
    % smooth transition to zero
    A(3*n+1,4*n-3) = 1;
    A(3*n+1,4*n-2) = x(n+1)-x(n);
    A(3*n+1,4*n-1) = (x(n+1)-x(n))^2;
    A(3*n+1,4*n) = (x(n+1)-x(n))^3;
    A(3*n+2,4*n-2) = 1;
    A(3*n+2,4*n-1) = 2*(x(n+1)-x(n));
    A(3*n+2,4*n) = 3*(x(n+1)-x(n))^2;
    A(3*n+3,4*n-1) = 2;
    A(3*n+3,4*n) = 6*(x(n+1)-x(n));

% add equations for the moments
for i=0:k-1
    for j=1:n
        aa = (x(j+1)^(i+1)-x(j)^(i+1))/(i+1);
        bb = (x(j+1)^(i+2)-x(j)^(i+2))/(i+2);
        cc = (x(j+1)^(i+3)-x(j)^(i+3))/(i+3);
        dd = (x(j+1)^(i+4)-x(j)^(i+4))/(i+4);
        sum = 0;
        for l=0:i
            sum = sum + x(j+1)^l*x(j)^(i-l);

```

```

end
aa = (x(j+1)-x(j)) * sum/(i+1);
sum = 0;
for l=0:i+1
    sum = sum + x(j+1)^l*x(j)^(i+1-l);
end
bb = (x(j+1)-x(j)) * sum/(i+2);
sum = 0;
for l=0:i+2
    sum = sum + x(j+1)^l*x(j)^(i+2-l);
end
cc = (x(j+1)-x(j)) * sum/(i+3);
sum = 0;
for l=0:i+3
    sum = sum + x(j+1)^l*x(j)^(i+3-l);
end
dd = (x(j+1)-x(j)) * sum/(i+4);
A(3*n+4+i,4*j-3) = aa;
A(3*n+4+i,4*j-2) = bb-x(j)*aa;
A(3*n+4+i,4*j-1) = cc-2*x(j)*bb+x(j)^2*aa;
A(3*n+4+i,4*j) = dd-3*x(j)*cc+3*x(j)^2*bb-x(j)^3*aa;
end
end

% read the values of the moments
[mom,xf,ff] = feval(example,[a/scale,b/scale],k);

% compute right hand side
% normalize the moments according to the scaling
norm_moment = mom(1)*scale;

% scale right hand side
for i=1:k
    rhs(s*n+s+i) = mom(i)*(scale^i)/norm_moment;
end

for j=0:n-1

    % do nothing for continuity
    % multiply with omega for C^1
    if s>1
        A(s*j+2,:) = A(s*j+2,)*omega;
    end

    % multiply with omega^2 for C^2
    if s==3
        A(3*j+3,:) = A(3*j+3,)*omega*omega;
    end
end

```

```

end

end

% Multiply both sides by the same factor to avoid deal with number with
% big difference
for i=1:k
    A(s*n+s+i,:) = A(s*n+s+i,:)*10^(-i+1);
    rhs(s*n+s+i) = rhs(s*n+s+i)*10^(-i+1);
end

% prepare solution of the linear system
tol = 1e-36;

% tolerance for continuity
cont_tol = 1e-2;
dis_cont = 0;

% loop over tolerance in SVD
while 1
    clf
    % flag for doing another iteration
    next = 0;
    % smallest value at the nodes
    value = 0;
    % largest value at the nodes
    max_value = 0;

    % solve linear system
    sol = pinv(A,tol)*rhs;

    % plot computed reconstruction
    step = 10;
    all_count = 1;

    % Number of iterations
    number = number + 1;
    mom_comp = zeros(k,1);

    hold on;

    % To calculate function values
    for j=1:n
        h = (x(j+1) - x(j))/step;
        for i=1:step+1
            xx(i) = x(j)+(i-1)*h;
            x_all(all_count) = xx(i)/scale;

```

```

% Function
fx(i) = sol(4*j-3) + sol(4*j-2)* (xx(i) - x(j)) + ...
sol(4*j-1)* (xx(i) - x(j))^2 + sol(4*j)* (xx(i) - x(j))
^3;

% First derivative
ffx(i) = sol(4*j-2) + ...
2*sol(4*j-1)* (xx(i) - x(j)) + 3*sol(4*j)* (xx(i) - x(j)
))^2;

% Second derivative
fffx(i) = 2*sol(4*j-1) + 6*sol(4*j)* (xx(i) - x(j));

f_all(all_count) = fx(i)*norm_moment;
all_count = all_count + 1;

% compute moments of the reconstructed function
if i>1
    val = (fx(i)+fx(i-1))*h/2;
    for l=1:k
        mom_comp(l) = mom_comp(l) + val;
        val = val * (xx(i)+xx(i-1))/2;
    end
end

% Maximum value of the function
if abs(fx(i))>max_value
    max_value = abs(fx(i));
end

% check continuity
if (j>1) && (i == 1)
    if abs(fx(1)-fx(step+1))>dis_cont
        dis_cont = abs(fx(1)-fx(step+1));
    end
end

end

% check of (strongly) negative values at the nodes
% if yes, than set the flag for a next iteration to 1
if fx(1) < value
    value = fx(1);
end

```

```

    if fx(6) < value
        value = fx(6);
    end

    % check if the domain of reconstruction can be made smaller
    % check if first interval is almost constant to zero
    if j == 1
        norm1 = norm(fx,inf);
    end

    % check if last interval is almost constant to zero
    if j == n
        norm2 = norm(fx,inf);
    end

    % plot all functions
    plot(xx(:)/scale,fx(:)*norm_moment,'k-');
    hold on
    plot(xx(1)/scale,fx(1)*norm_moment,'ko',xx(11)/scale,fx(11)*
        norm_moment,'bo');
    hold on

    % save data for the adaptive grid
    % save data for plot
    xdata(j*(step+1)-step:j*(step+1),auto_regrid)=(xx/scale)';
    ydata(j*(step+1)-step:j*(step+1),auto_regrid)=(fx*norm_moment)';
    xx1data(j*(step+1)-step:auto_regrid)=(xx(1)/scale)';
    yy1data(j*(step+1)-step:auto_regrid)=(fx(1)*norm_moment)';
    xx11data(j*(step+1)-step:auto_regrid)=(xx(11)/scale)';
    yy11data(j*(step+1)-step:auto_regrid)=(fx(11)*norm_moment)';
    % save first derivative
    dy((j-1)*step+1:j*(step+1)-(j-1)*1,1)=ffx';
    % save second derivative
    ddy((j-1)*step+1:j*(step+1)-(j-1)*1,1)=fffx';
    % save x data
    ddx((j-1)*step+1:j*(step+1)-(j-1)*1,1)=xx';

end

%
hold on
st=sprintf('reconstruction %d',number);
title(st);

%legend('function','reconstruction','Location','Best');
%legend('PSD','reconstruction','nodes');
legend('reconstruction','nodes');

```



```

xlabel('size, [m]');
ylabel('distribution, [1/m]');
pause(1e-2)

% check if interval can be decreased
max_value_vec(number) = max_value;
max_value = norm(max_value_vec, -inf);
value = value/max_value_vec(number);

if value < minimum
    next = 1;
end

% solution is discontinuous
if dis_cont/max_value_vec(number) > cont_tol
    break;
end

% check if the point most left can be moved to the right
if (norm1 < 1e-2*max_value)
    redistribute_left = redistribute_left + 1;
end

% check if the point most right can be moved to the left
if norm2 < 1e-2*max_value
    redistribute_right = redistribute_right + 1;
end

% check if nodes should be redistributed
% this is just for the seed example
redistribute_left = 0;
if redistribute_left == 1
    % set most left point to the centre of the first interval
    a = (x(1)+x(2))/2;
    redistribute_nodes = 1;
end

if redistribute_right == 1
    % set most right point to the centre of the last interval

    % if it is in the equidistant grid loop
    if auto_regrid == 1
        b = (x(n+1)+x(n))/2;
    end

    % if it is in the adaptive loop
    if auto_regrid > 1

```

```

        b = (0.75*x(n+1)+0.25*x(n))/1;
    end

    redistribute_nodes = 1;
end

% redistribute nodes according to the new boundaries of the
% interval, currently equidistant
if redistribute_nodes == 1

    % this is only done if it is in the equidistant grid loop
    if auto_regrid==1
        h = (b-a)/n;
        x(1) = a;
        x(n+1) = b;
        for l=2:n
            x(l) = a + (l-1)*h;
            % b_ value to be used to set the domain for the adaptive grid
            b_=b;
        end
    end

    break; % loop over tolerance

    % after a redistribution everything is started from the beginning
    % with the new interval
end

% no redistribution of the nodes was done
% check if all values are non-negative
if next > 0
    % neglect the next singular value
    format long e;
    svd_vector = svd(A);
    [l,m] = size(A);
    %[l,m] = size(x);
    while(svd_vector(l)<tol)
        l = l-1;
    end
    tol = (svd_vector(l)+svd_vector(l-1))/2;
else
    break; %loop over tolerance
end

end % loop over tolerance

% final reconstruction found
if redistribute_nodes ~= 1

```

```

        break;
    end

    % if the distance between the last nodes is getting to small
    if (x(n+1)-x(n)) < (b/(2*n))
        break;
    end
end % loop over redistribution

% calculate relative error in the moments
for i=1:k
    mom_comp(i) = mom_comp(i)*norm_moment/(scale^i);
end
mom_error(auto_regrid,:) = (((mom_comp - mom(1:k))' ./ mom(1:k)') * 100);

% Adaptive grid
% Mean error for the moments
mean_mom_error(auto_regrid) = mean(abs(mom_error(auto_regrid,:)));

if auto_regrid > 1

% Criteria to stop adaptive grid
if mean_mom_error(auto_regrid) >= mean_mom_error(auto_regrid-1) && max(abs(
    (mom_error(auto_regrid-1,:))) < 30
% plot the distribution
    close(1)
    figure(1)
    plot(xdata(:,auto_regrid-1), ydata(:,auto_regrid-1), 'k-', '
        linewidth', 2);
    hold on
    plot(xx1data(:,auto_regrid-1), yy1data(:,auto_regrid-1), 'ko', xx11
        data(:,auto_regrid-1), yy11data(:,auto_regrid-1), 'bo', '
        markerfacecolor', 'b', 'markersize', 8);
    hold on
    st = sprintf('reconstruction %d', number_);
    title(st);

    %legend('function', 'reconstruction', 'Location', 'Best');
    %legend('PSD', 'reconstruction', 'nodes');
    legend('reconstruction', 'nodes');
    %xlabel('crystal size, [m]');
    %ylabel('crystal size distribution, [1/m]');

    disp('relative error in moments');
    abs(mom_error(auto_regrid-1,:))

break;

```

```

else

end

end

% Number of iterations
number_=number;

% New domain
b=b_*b_domain;

% Second derivative avaliation
ddy = ddy(2:(step*n));
% Find points where signal is rising before, falling after
upordown = [ddy>0];
maxflags=[diff(upordown)<0];
max_ = find(maxflags);
% Find points where signal is falling before, rising after
minflags=[diff(upordown)>0];
min_ = find(minflags);
% Points where the second derivative changes the signal
x=[0;ddx(min_);ddx(max_)];

% First derivative avaliation
[mx,nx]=size(x);
% number of internal points that still should be chosen
nip=n-mx;
% total number of nodes minus the right boundary
npt=k+s;
% value of the maximum first derivative
dy=abs(dy);
maxdy=max(dy);
% Step to take different first derivative regions "windows"
d_maxdy=maxdy/(nip+1);

dx=[];
dx_=[];
dx_=[1;min_;max_];
dx_=sort(dx_);
[mdx,ndx]=size(dx_);

for j=1:(nip+1)

    dyindex_=[];

```

```

% The first derivative "window"
index=[dy<(maxdy+1e-36) & dy>((maxdy+1e-36)-d_maxdy)];
% Find the points
dyindex_=find(index);
% Put points of second and first derivative together
dx=[dx_;dyindex_];

% Check if adjacent points have the minimum distance allowed
[mx,nx]=size(dx);
for i=1:mx-2
    if abs(ddx(dx(mx+1-i))-ddx(dx(mx-i)))<(b/min_space)
        dx(mx-i)=[];
    end
end

for i=mx-1
    if abs(ddx(dx(mx+1-i))-ddx(dx(mx-i)))<(b/min_space)
        dx(mx+1-i)=[];
    end
end

% Delete points in excess
[mx,nx]=size(dx);
if mx>npt
    dx(npt+1:mx)=[];
end

% delete points in the cluster
dx=sort(dx);
[mx,nx]=size(dx);
for i=1:mx-2
    if (ddx(dx(mx+1-i))-ddx(dx(mx-i)))<(b/min_space)
        dx(mx-i)=[];
    end
end

% Check if adjacent points have the minimum distance allowed
for i=mx-1
    if abs(ddx(dx(mx+1-i))-ddx(dx(mx-i)))<(b/min_space)
        dx(mx+1-i)=[];
    end
end

[mx,nx]=size(dx);
dx_=dx;

% if the correct number of points were chosen

```

```

    if mx==npt
    break;
end

    % For the next first derivative "window"
    maxdy=(maxdy)-d_maxdy;

end

% The new position of the nodes
x=[ddx(dx)];
x(n+1)=b;
% To organize the points in ascendent order
x=sort(x);
% Change auto adaptive flag for next iteration
auto_regrid=auto_regrid+1;

end
toc
save_data_grafic = [xdata(:,auto_regrid-1)*100,ydata(:,auto_regrid-1)/10
    0];
save Spline10 .mat save_data_grafic

```

Arquivo que contém o vetor de momentos:

```

function [mu,xf,ff] = moments(a,k)
% values computed with the method of moments

mu(1) = 1.00;
mu(2) = 2.64055e+02*1.0e-2;
mu(3) = 8.11618e+04*1.0e-4;
mu(4) = 2.88274e+07*1.0e-6;
mu(5) = 1.16942e+10*1.0e-8;
mu(6) = 5.33360906514972e+12*1.0e-10;
mu(7) = 2.68615963646757e+15*1.0e-12;
mu(8) = 1.46711536956573e+018*1.0e-14;
mu(9) = 8.54941312261519e+020*1.0e-16;
mu(10) = 5.24334002231077e+023*1.0e-18;

for i=k+1:40
    mu(i) = 0;
end
h = (a(2)-a(1))/1000;
for i=1:1001
    xf(i) = a(1)+(i-1)*h;
    ff(i) = 0;
end
endfunction

```

ANEXO B – SUB-ROTINA PARA CÁLCULO DE AUTOVALORES E AUTOVETORES

Essa sub-rotina foi utilizada para realizar o calculo de autovalores e autovetores do algoritmo Produto-Diferença.

```

subroutine jacobi_eigenvalue ( n, a, it_max, v, d, it_num,
&    rot_num )

!*****
!
! JACOBI_EIGENVALUE carries out the Jacobi eigenvalue iteration.
!
! Discussion:
!
! This function computes the eigenvalues and eigenvectors of a
! real symmetric matrix, using Rutishauser modifications of the classical
! Jacobi rotation method with threshold pivoting.
!
! Licensing:
!
! This code is distributed under the GNU LGPL license.
!
! Modified:
!
! 17 September 2013
!
! Author:
!
! FORTRAN77 version by John Burkardt
!
! Parameters:
!
! Input, integer N, the order of the matrix.
!
! Input, double precision A(N,N), the matrix, which must be square, real,
! and symmetric.
!
! Input, integer IT_MAX, the maximum number of iterations.
!

```

```
! Output, double precision V(N,N), the matrix of eigenvectors.
!
! Output, double precision D(N), the eigenvalues, in descending order.
!
! Output, integer IT_NUM, the total number of iterations.
!
! Output, integer ROT_NUM, the total number of rotations.
!

  implicit none

  integer n

  double precision a(n,n)
  double precision bw(n)
  double precision c
  double precision d(n)
  double precision g
  double precision gapq
  double precision h
  integer i
  integer it_max
  integer it_num
  integer j
  integer k
  integer l
  integer m
  integer p
  integer q
  integer rot_num
  double precision s
  double precision t
  double precision tau
  double precision term
  double precision termp
  double precision termq
  double precision theta
  double precision thresh
  double precision v(n,n)
  double precision w(n)
  double precision zw(n)

  do j = 1, n
    do i = 1, n
      v(i,j) = 0.0D+00
    end do
    v(j,j) = 1.0D+00
  end do
```



```

do i = 1, n
    d(i) = a(i,i)
end do

do i = 1, n
    bw(i) = d(i)
    zw(i) = 0.0D+00
end do

it_num = 0
rot_num = 0

10 continue

if ( it_num .lt. it_max ) then

    it_num = it_num + 1

!
! The convergence threshold is based on the size of the elements in
! the strict upper triangle of the matrix.
!

    thresh = 0.0D+00
    do j = 1, n
        do i = 1, j - 1
            thresh = thresh + a(i,j) ** 2
        end do
    end do

    thresh = sqrt ( thresh ) / dble ( 4 * n )

    if ( thresh .eq. 0.0D+00 ) then
        go to 20
    end if

    do p = 1, n
        do q = p + 1, n

            gapq = 10.0D+00 * abs ( a(p,q) )
            termq = gapq + abs ( d(p) )
            termq = gapq + abs ( d(q) )

!
! Annihilate tiny offdiagonal elements.
!

            if ( 4 .lt. it_num .and.
&                termq .eq. abs ( d(p) ) .and.
&                termq .eq. abs ( d(q) ) ) then

```

```

        a(p,q) = 0.0D+00
!
! Otherwise, apply a rotation.
!
        else if ( thresh .le. abs ( a(p,q) ) ) then

            h = d(q) - d(p)
            term = abs ( h ) + gapq

            if ( term .eq. abs ( h ) ) then
                t = a(p,q) / h
            else
                theta = 0.5D+00 * h / a(p,q)
                t = 1.0D+00 /
&                ( abs ( theta ) + sqrt ( 1.0D+00 + theta * theta ) )
                if ( theta .lt. 0.0D+00 ) then
                    t = - t
                end if
            end if

            c = 1.0D+00 / sqrt ( 1.0D+00 + t * t )
            s = t * c
            tau = s / ( 1.0D+00 + c )
            h = t * a(p,q)

!
! Accumulate corrections to diagonal elements.
!

            zw(p) = zw(p) - h
            zw(q) = zw(q) + h
            d(p) = d(p) - h
            d(q) = d(q) + h

            a(p,q) = 0.0D+00
!
! Rotate, using information from the upper triangle of A only.
!

            do j = 1, p - 1
                g = a(j,p)
                h = a(j,q)
                a(j,p) = g - s * ( h + g * tau )
                a(j,q) = h + s * ( g - h * tau )
            end do

            do j = p + 1, q - 1
                g = a(p,j)
                h = a(j,q)

```

```

        a(p,j) = g - s * ( h + g * tau )
        a(j,q) = h + s * ( g - h * tau )
    end do

    do j = q + 1, n
        g = a(p,j)
        h = a(q,j)
        a(p,j) = g - s * ( h + g * tau )
        a(q,j) = h + s * ( g - h * tau )
    end do

!
! Accumulate information in the eigenvector matrix.
!

    do j = 1, n
        g = v(j,p)
        h = v(j,q)
        v(j,p) = g - s * ( h + g * tau )
        v(j,q) = h + s * ( g - h * tau )
    end do

    rot_num = rot_num + 1

    end if

    end do
end do

do i = 1, n
    bw(i) = bw(i) + zw(i)
    d(i) = bw(i)
    zw(i) = 0.0D+00
end do

go to 10

end if

20    continue
!
! Restore upper triangle of input matrix.
!

    do j = 1, n
        do i = 1, j - 1
            a(i,j) = a(j,i)
        end do
    end do

!
```

```
! Ascending sort the eigenvalues and eigenvectors.
!  
do k = 1, n - 1  
  
    m = k  
  
    do l = k + 1, n  
        if ( d(l) .lt. d(m) ) then  
            m = l  
        end if  
    end do  
  
    if ( m .ne. k ) then  
  
        t      = d(m)  
        d(m) = d(k)  
        d(k) = t  
  
        do i = 1, n  
            w(i) = v(i,m)  
            v(i,m) = v(i,k)  
            v(i,k) = w(i)  
        end do  
  
    end if  
  
end do  
  
return
```